Aalto-yliopisto
Sähkötekniikan korkeakoulu
Automaatio- ja sähkötekniikan DI-ohjelma

Mikko Rekola

Panostuksen seuranta:
Emulsiopanostus tunneliolosuhteissa

Diplomityö
Espoo, 15. marraskuuta 2017

Valvoja: Prof. Arto Visala
Ohjaaja: DI Saku Viita-Aho

Aalto-yliopisto
Sähkötekniikan korkeakoulu
Automaatio- ja sähkötekniikan DI-ohjelma

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä:	Mikko Rekola		
Työn nimi:	Panostuksen seuranta: Emulsiopanostus tunneliolosuhteissa		
Päiväys:	15. marraskuuta 2017	**Sivumäärä:**	vii + 61
Professuuri:	Automaatiotekniikka	**Koodi:**	AS-84
Valvoja:	Prof. Arto Visala		
Ohjaaja:	DI Saku Viita-Aho		

Tässä diplomityössä selvitetään, mikä on paras tapa seurata emulsiopanostusprosessin etenemistä tunneliolosuhteissa. Erityisesti ollaan kiinnostuneita seuraamaan sitä, kuinka paljon emulsioräjähdysainetta pumpataan kuhunkin panostettavaan reikään. Työssä analysoidaan tunneliympäristön asettamia rajoitteita ja esitellään vaihtoehtoisia menetelmiä ongelman ratkaisemiseksi.

Seurantaa varten rakennettiin prototyyppi, joka etsii pistepilvestä panostusletkuksi pääteltäviä sylinterimuotoja. Letkuntunnistusalgoritmi toteutettiin pääosin RANSAC algoritmilla. Prototyypin perusteella havaittiin, että pistepilvien muodostuksessa käytetty Kinect 2.0 kamera ei ole tarpeeksi tarkka sylintereiden tarkkarajaiseen löytämiseen. Toinen havaittu ongelma on kuvausalueen pienuus, minkä vuoksi toimiva toteutus vaatii useampia kameroita.

Työn tulos on, että käsin kirjaaminen on tehokkain tapa seurata panostusprosessin etenemistä johtuen muiden menetelmien epävarmuudesta. Jatkotutkimusvaihtoehtoina esitetään toimivan kamerakonfiguraation etsimistä, referenssikuvan muodostusta liikkuvan kameran avulla sekä luotettavuuden parantamista usean seurantamenetelmän yhteistyöllä.

Asiasanat:	panostus, emulsiot, seuranta, konenäkö, pistepilvet, RANSAC, maanalaiset olosuhteet, räjäytys, sensorit, 3D
Kieli:	Suomi

Aalto University
School of Electrical Engineering
Master's Programme in Automation and Electrical Engineering

ABSTRACT OF MASTER'S THESIS

Author:	Mikko Rekola		
Title:	Excavation Charging Monitoring: Emulsion Charging in Tunnel Environment		
Date:	November 15th, 2017	**Pages:**	vii + 61
Professorship:	Automation Technology	**Code:**	AS-84
Supervisor:	Prof. Arto Visala		
Advisor:	M.Sc. (Tech.) Saku Viita-Aho		

This Master's Thesis examines the best way to monitor the progress of an emulsion charging process in tunnel environment. In particular, the point of interest is how much explosive emulsion is pumped into each hole to be charged. The thesis analyzes the constraints imposed by the tunnel environment and presents alternative methods to solve the problem.

A prototype which looks for cylinder shapes from the point cloud was constructed for monitoring. The algorithm for hose detection was implemented with the RANSAC algorithm. Based on the prototype, it was found that the Kinect 2.0 camera used in the formation of the point clouds is not accurate enough to detect the cylinders exactly. Another problem encountered was the small field of view of the camera, which means that many cameras are required in the implementation.

The result of the thesis is that manual logging is the most effective way to monitor the progress of the charging process. This is due to the uncertainty of other methods. Further research options include finding a working camera configuration, forming a reference point cloud using a moving camera and improving reliability by combining multiple monitoring methods.

Keywords:	charging, emulsions, monitoring, machine vision, point clouds, RANSAC, underground environment, blasting, sensors, 3D
Language:	Finnish

Esipuhe

Tämä diplomityö tehtiin YIT:n tilauksesta Viasys VDC:n tuottamana. Työ lähti liikkeelle jo kaksi vuotta sitten, kun ensimmäisen kerran kokoonnuimme silloisen Vianovan kokoustilassa Ville Hervan ja Tarmo Räntilän kanssa. Minä ja Tarmo tarvitsimme diplomityöaiheet ja esimiehemme Ville oli ne lupautunut järjestämään. Sopivien aiheiden löytämiseen meni noin puoli vuotta, jonka jälkeen YIT:n kanssa tehtiin sopimus kahden diplomityön toteuttamisesta. En tällöin vielä arvannut, kuinka rankka kokemus työ tulisi olemaan.

Tekemisen ensimmäiset askeleet olivat toiveikkaita, mutta pian eteneminen pysähtyi. Viimeistään kesän koittaessa muut työt kiilasivat edelle, jolloin diplomityö jäi taka-alalle. Syksyllä otin aikaa diplomityön tekemiselle, mutta prototyyppitoteutuksen ongelmat ja tästä seurannut prokrastinointi estivät todellisen tuloksen syntymisen. Tämä muuttui vasta seuraavana keväänä, jolloin sain prototyypin toimimaan jollakin tasolla. Tämän jälkeen kirjoitustyö eteni muiden töiden ohessa tähän päivään asti.

En varmastikaan olisi pystynyt saattamaan työtä loppuun täysin ilman apua. Suurimmat kiitokset kuuluvat Tarmolle, jonka kanssa keskustellessa useat työhön liittyvät ongelmat ratkesivat. Yhtä tärkeässä roolissa oli Silja-Maria Rytkönen, joka auttoi jaksamaan kirjoitusprosessin läpi ja löysi lukuisat kirjoitusvirheeni. Kiitokset myös Ville Hervalle ja työn lopulliselle ohjaajalle Saku Viita-Aholle, sekä työn valvojalle Arto Visalalle. Kiitokset tietysti myös vanhemmilleni, jotka olivat aina valmiita auttamaan arjen ongelmissa.

Espoo, 15. marraskuuta 2017

Mikko Rekola

Lyhenteet ja käsitteet

Emulsio	Toisiinsa luonnostaan sekoittumattomien nesteiden seos
ICP	Iterative Closest Point -algoritmi
IMU	Inertial Measurement Unit
Kd-puu	Binäärinen etsintäpuu
PCL	PointCloud Library
Pistepilvi	Joukko 3D-pisteitä
Poraus-räjäytyskaavio	Kuvio jonka mukaan tunneliin porataan panostettavat reiät
RANSAC	Iteratiivinen parametrinestimointimenetelmä

Sisältö

Luku 1

Johdanto

Tunnelien rakentaminen on paljon muutakin, kuin vain kuopan kaivamista maahan. Merkittävä osa tunneleista rakennetaan peruskallioon ja nykyaikana tehokkain tapa on kiviaineksen räjäyttäminen ja pois kuljettaminen. Tunnelirakentamiseen sisältyy useita työvaiheita, jotka täytyy suorittaa peräjälkeen tarkoin aikataulutetussa prosessissa. Siinä on tärkeää, että työmaa pysyy aikataulussa ja että työvaiheet etenevät oikeaan aikaan. Seuraavaa työvaihetta ei voida aloittaa ennen kuin edellinen tehtävä on suoritettu. Räjäyttää ei voi ennen panostusta, panostaa ei voi poraamatta reikiä ja reikiä ei voi porata, ellei edellinen räjäytys ole suoritettu, tunneli tuuletettu ja kiviaines kuljetettu pois [1]. Yhden työvaiheen viivästyminen tarkoittaa, että seuraavaan työvaiheeseen varatut resurssit joutuvat kaikki odottamaan. Aikataulun pitäminen on kriittistä varsinkin iltaisin kohteissa, joissa räjäytysaika on rajoitettu. Tällaisia ovat esimerkiksi asuinalueiden läheisyydessä sijaitsevat työmaat, joissa räjäytyksessä syntyvä melu häiritsee merkittävissä määrin ympäristön asukkaita, eikä räjäytyksiä siten voida suorittaa yöllä [2]. Pienemmissä kohteissa odotteluaikana ei välttämättä voi tehdä muuta hyödyllistä, mikä aiheuttaa kustannuksia rakennusyhtiölle. Pahimmillaan viivästyminen voi aiheuttaa jopa vuorokauden pysähdyksen työmaalla.

Asuinalueella räjäyttäminen aiheuttaa erityisiä ongelmia tunnelirakentamiselle. Vain 20-30% räjäytykseen käytetystä energiasta kuluu kiven rikkomiseen [3]. Muu osuus kulkee värähtelynä kaikkialle ympäristöön. Räjäytyksen voimaa voidaan hillitä jakamalla se useampiin peräkkäin räjähtäviin panoksiin [1]. Näin räjähdyksen energia jakaantuu pidemmälle aikavälille, mikä vähentää havaittavaa värähtelyä maan pinnalla. Räjäytysprosessi on tarkasti määritelty ja tiedetään myös suhteellisen tarkasti, paljonko räjähdettä tarvitsee käyttää halutun tuloksen saamiseen. Räjähdeaineen määrää ei kuitenkaan tarkasti pystytä seuraamaan, vaan sitä rajoittaa panostetun reiän koko ja panostajan ammattitaito.

Räjäytystyötä määrittää useat lait ja säädökset. Ne takaavat oikein noudatettuna sekä ympäristön, että räjäytystyön tekijän turvallisuuden. Räjäytystyön kohdalla merkittäviä asetuksia ovat "Valtioneuvoston asetus räjähteiden valmistuksen, käsittelyn ja varastoinnin turvallisuusvaatimuksista (1101/2015)", sekä "Valtioneuvoston asetus räjäytys- ja louhintatyön turvallisuudesta (644/2011)" [4][5]. Näiden säädösten seuraaminen voi olla joillekin yrityksille hankalaa ja joissakin tapauksissa jopa teknisesti haastavaa. Yksi tällainen on asetus räjäytys- ja louhintatyön turvallisuudesta § 12, joka määrää: "Työmaalle tuodusta, käytetystä ja luovutetusta räjähteestä on pidettävä kirjaa." Käytännössä tämä tarkoittaa, että yrityksen on seurattava panostuksessa sitä, kuinka paljon räjähdeainetta menee kuhunkin porattuun reikään.

Nykyisellään tiedetään, kuinka paljon räjähdeainetta menee reikään yhdellä panostuksella, mutta ei sitä missä järjestyksessä panostaja on reiät panostanut. Jälkikäteen ei voida tietää, oliko rei'issä oikeaa määrä räjähdeainetta. Kuitenkin yrityksen tulee seurata tätä, mikä on nykyisellään mahdotonta aiheuttamatta huomattavaa haittaa panostusprosessille. Maailmalla on tehty useita tutkimuksia räjäytystöistä. Ne keskittyvät kuitenkin pääasiassa itse räjäytykseen ja erilaisten menetelmien ympäristövaikutuksiin [6][7][8]. Varsinaisen panostusprosessin seuraamista ei ole tehty siis juuri lainkaan.

1.1 Työn tavoite

Tämän diplomityön tarkoituksena on selvittää paras keino seurata panostajan toimintaa panostustilanteessa. Tarkemmin selvitetään sitä, missä järjestyksessä ja kuinka paljon panostaja pistää räjähdysainetta kuhunkin panostettavaan reikään. Työssä keskitytään erityisesti tunneliympäristöön ja tunnelinperän panostamiseen. Lähtökohtaisesti tiedetään, kuinka paljon räjähdeainetta kuluu kun panostaja laittaa räjähde ainetta reikään. Sitä puolestaan ei seurata toistaiseksi millään tavalla, mihin reikään räjähteet on pistetty. Panostuksessa kuluneen räjähdeaineen määrän lisäksi tunnelinperän räjäytyskuvio tunnetaan tarkasti. Selvitystyön ohessa toteutetaan prototyyppi, jonka avulla tutkitaan tarkemmin valitun menetelmän toimivuutta panostuksen seurantaan. Erityistarkastelussa ovat tunneliympäristön vaikutukset ongelman ratkaisuun, sekä panostusprosessin erityispiirteet. Lopputuloksena toimitetaan prototyyppitoteutus, sekä esitetään näkemyksiä jatkotoimista.

Tämä diplomityö toteutettiin YIT:n tilauksesta Viasys VDC:n toimesta. YIT:n lähtökohta tutkimukselle oli, että se itse ei tee panostusprosessin seu-

rannan tuloksilla mitään, mutta sen täytyy asetuksen mukaan sitä seurata. Käytännössä kuitenkin olisi mahdollista, että menetelmällä voisi olla tulevaisuudessa muita käyttökohteita, esimerkiksi panostuksen jälkiseurantaan tapauksessa, jossa räjäytyksen lopputulos ei olisi toivottu. Varsinaiseen panostusprosessiin millä tahansa toteutetulla menetelmällä on todennäköisesti ainoastaan haittaava vaikutus, joten jos sopiva menetelmä löytyy, ei se saa haitata panostusprosessia liiiaksi.

1.2 Diplomityön rakenne

Ensimmäisenä työssä tarkastellaan panostusprosessia kokonaisuutena. Tähän sisältyy tunneliympäristön asettamat rajoitteet erilaisille menetelmille ja potentiaalisten laitteistojen ominaisuuksille. Toisessa osassa käsitellään erilaisia ratkaisuvaihtoehtoja ja pohditaan niiden käytännöllisyyttä ongelman kannalta sekä toteutuskelpoisuutta. Kolmannessa osassa tarkastellaan tarkemmin valittua menetelmää, eli pistepilvianalyysin teoriaa.

Työn kokeellisessa osuudessa esitellään tuotettu prototyyppi. Osuudessa kerrotaan mihin vaiheeseen prototyyppi saatiin toteutettua ja mitä siihen vielä vaadittaisiin, jotta se olisi toimiva tuote. Lopulliseksi tuotteeksi määritellään systeemi, joka seuraa itsenäisesti panostajan toimintaa, analysoi tiedot ja yhdistää ne olemassa olevaan tietoon panostuksesta ja räjähdysaineiden määrästä. Erityistä huomiota tässä osassa kiinnitetään siihen, että tuotettu laite ei merkittävästi häiritse panostusprosessia. Lopulta analysoidaan, kuinka hyvin tuotettu prototyyppi täyttää sille asetetut vaatimukset ja toimii käytännön olosuhteissa. Tässä esitetään myös arvio valitusta menetelmästä ja siitä, kannattaako kyseisen menetelmän kehittämistä jatkaa.

Luku 2

Tausta

Tunneleita rakennetaan nykyään pääasiassa räjäyttämällä kaistale kerrallaan tunnelin päästä. Yhtä tällaista kaistaletta kutsutaan katkoksi [9]. Yhden katkon rakentamisessa on useita työvaiheita, mutta tämän työn tutkimuskohde liittyy pääasiassa panostusvaiheeseen. YIT:llä on tarve selvittää, kuinka paljon panostukseen käytettävää emulsioräjähdettä kuluu jokaista panostettua reikää kohden. On tiedossa, kuinka paljon räjähdysainetta on kulunut panostusprosessissa yhteensä, mutta reikäkohtaista seurantaa ei ole toistaiseksi ollut mahdollista toteuttaa. Pääasiallinen tavoite onkin selvittää, millä tavalla se voisi olla mahdollista. Pelkkä teoreettinen tarkastelu ei tietenkään riitä, vaan ongelmaa täytyy tarkastella käyttäjälähtöisesti. Ratkaisu ei saa aiheuttaa liikaa ylimääräisiä kustannuksia ja menetelmän täytyy olla luotettava. Ongelmaan pureutumiseksi tässä luvussa käydään läpi tarvittavaa taustatietoa liittyen tunneleiden louhimiseen, räjäytyksiin ja tunneliympäristön erityispiirteisiin verrattuna maanpäälliseen toimintaan. Ensimmäisenä käsitellään tunnelilouhinnan perusteet. Toisena tarkkaillaan nimenomaisesti panostusprosessia. Viimeisenä havainnoidaan tunneliympäristöä ja sen rajoitteita.

2.1 Tunnelirakentaminen

Tunneleita louhitaan yleensä sen takia, että maan päällä ei ole enää tilaa rakentaa tai maan päälle rakentaminen on maaston muodoista johtuen haastavaa ja kallista verrattuna tunnelin rakentamiseen. Kaupunkiympäristössä tilan puute on merkittävin tekijä. Maan alle rakennetaan tällöin liikenneyhteyksiä tai parkkihalleja, mikä mahdollistaa lyhyet etäisyydet maan pinnalla oleviin pääasiallisiin kohteisiin. Luonnossa puolestaan tunneleita rakennetaan hankalien maastonmuotojen ohittamiseen. Tyypillisesti tämä tarkoit-

taa mäen tai vuoren alitusta, mutta tunneli voidaan rakentaa myös alittamaan vesialue. Kolmas syy tunnelin rakentamiselle on tarve päästä maan alle. Maan alla voi olla arvokkaita mineraaleja, minkä vuoksi esiintymän päälle halutaan rakentaa kaivos. Joissakin tapauksissa maan alla on paremmat olosuhteet tiettyyn tarpeeseen kuin maan päällä. Maanalaiset luolat tarjoavat suojaa maan päällä tapahtuvilta mullistuksilta. Niissä on myös usein viileämpää, mikä tekee niistä käytännöllisiä varastotiloina. [10]

Historiallisesti tunneleiden rakentaminen oli käsityötä, joka tapahtui hakuin ja lapioin. Nykyään tärkein työväline ovat räjähteet. Niitä käytettiin louhintatöissä ensimmäisen kerran jo vuonna 1627. Kaivostoiminnassa niitä alettiin hyödyntää kuitenkin vasta vuonna 1811 puhtaan trotyylin, eli TNT:n keksimisen jälkeen. Tästä alkoi nopea räjähdysaineiden kehitys, joka on jatkunut tähän päivään asti. Tänä aikana ovat kehittyneet niin räjähteet, sytytysjärjestelmät kuin räjäytystekniikatkin. Porausmenetelmissä käännekohta tapahtui 1960-luvulla, kun siirryttiin käsiporauksesta laitteistoihin, joilla yksi henkilö pystyi porata useampaa reikää samaan aikaan. Vuonna 2005 pystyttiin suurimmilla laitteilla porata jo 450 porametriä tunnissa. Erityisesti on parantunut turvallisuus: vielä 1950-luvulla Suomessa kuoli räjäytystöissä ja varastoinnissa yhteensä 52 henkilöä. 1990-luvulla tämä määrä oli laskenut jo kolmeen henkilöön. Tekniikan kehittymisen lisäksi turvallisuuteen ovat vaikuttaneet lait ja määräykset, joiden vuoksi tätäkin työtä lähdettiin tekemään. Vasta vuonna 1993 poistettiin jokamiehenoikeus tehdä vähäisiä räjäytystöitä, ja kaikkiin räjäytystöihin tarvitsee nykyään olla panostajan lupakirja. Louhintaräjähdysaineiden käytön kasvulle ei ole näkymässä loppua, joten louhintatekniikoiden kehittymistä voidaan odottaa tulevaisuudessakin. Vuonna 1965 oli räjähdysaineena käytössä vielä enimmäkseen dynamiitti. Nykyään tässä työssä käsiteltävät bulk-emulsiot ovat vallanneet suurimman osan räjähdeainekentästä ja ne ovat Vuolion mukaan myös tulevaisuuden räjähdysaine. [11]

Tunnelirakentaminen on iteratiivinen prosessi. Suomessa ainoa käytössä oleva menetelmä maanalaiseen louhintaan on poraus-räjäytysmenetelmä, jossa louhinta tapahtuu vaihe kerrallaan kuvan 2.1 mukaisesti. Porauksessa räjäytettävään seinämään tehdään poraus-räjäytyskaavion mukainen kuvio reikiä. Porattujen reikien pituus vaihtelee suunnitellun katkon mukaan. Yhden katkon pituus voi vaihdella puolesta metristä kuuteen metriin asti [9]. Panostuksessa porattuihin reikiin sijoitetaan räjäytysaine. Tätä käsitellään seuraavassa kappaleessa tarkemmin. Räjäytysvaiheessa panokset räjäytetään, jonka jälkeen tunneli täytyy tuulettaa, jotta räjähdyksessä syntyneet kaasut saadaan poistettua ja seuraava työvaihe aloitettua. Kuormauksessa ja kuljetuksessa louhe poistetaan, jonka jälkeen suoritetaan rusnaus, lujitus ja injektointi tarpeen mukaan. Rusnauksessa irrotetaan tunnelin seinistä ja katosta

Kuva 2.1: Poraus-räjäytysmenetelmän työvaiheet: poraus, panostus ja kytkennät, räjäytys, tuuletus, kuormaus, kuljetus, rusnaus, lujitus ja injektointi [12]

helposti irtoavat kappaleet. Lujituksessa ja injektoinnissa puolestaan seinämä vahvistetaan pulteilla ja betonoinnilla, jotta tunnelissa on turvallista liikkua. Tämän jälkeen prosessi alkaa uudestaan porauksella. [12]

Räjähteitä käytettäessä on olemassa neljä pääasiallista räjäytystekniikkaa: linjaporaus, esihalkaisu, peiteräjäytys ja sileä räjäytys. Näistä ainoastaan sileää räjäyttämistä käytetään maan alla, sillä muut ovat pintaräjäytystekniikoita [13]. Sileä räjäyttäminen tarkoittaa, että pyritään räjäyttämään tunneli valmiiksi juuri oikeaan muotoon. Tällöin ei tarvitse käyttää aikaa ja resursseja räjäytyksen epätasaisuuksien siivoamiseen. Sileä räjäyttäminen toteutetaan käyttämällä suhteellisen pieniä panoksia järjestettynä optimoituun räjäytyskuvioon [14]. Liian suurella panoksella tunnelin seinämiin ja kattoon syntyy koloja, jotka joudutaan betonoimaan umpeen. Toisaalta liian pieni panos jättää tunnelin liian kapeaksi, mikä joudutaan korjaamaan koneellisesti. Tämä on yksi tekijä, jonka takia panostusta on hyvä kyetä seuraamaan.

2.2 Panostusprosessi

Diplomityön tutkimusongelma liittyy keskeisesti panostusprosessiin, joten se käydään läpi tarkemmin. Ennen panostusta tunnelinperään on porattu reikiä tunnetun räjäytyskuvion mukaan. Panostuksessa näihin reikiin asetetaan räjähteet sekä kytkennät. Panostuksessa käytetty räjähdysaine voidaan valita kolmen yleisesti käytetyn vaihtoehdon joukosta. Ensimmäinen vaihtoehto on emulsioräjähde.

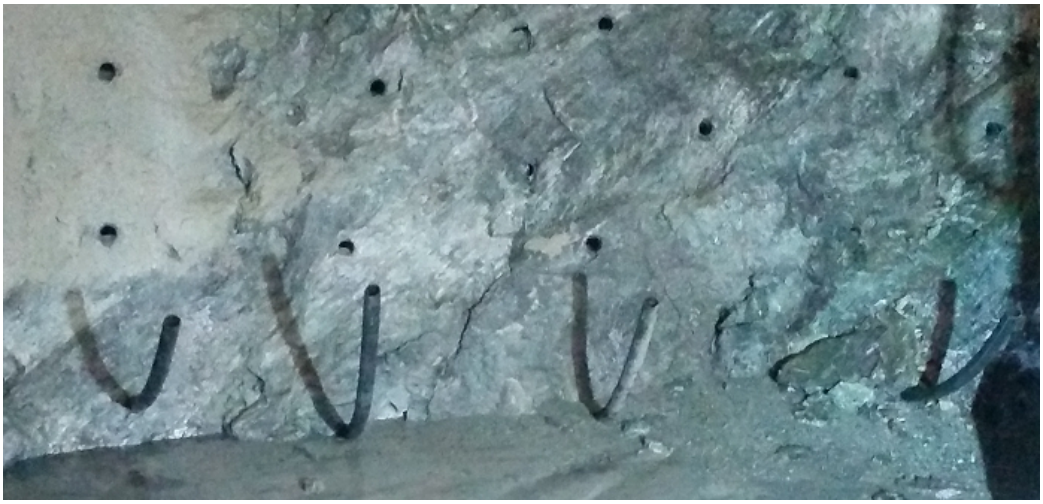
Emulsiot ovat geelimäisiä räjähdysaineita, jotka koostuvat matriisiosasta sekä kaasutusliuoksesta. Emulsioita on sekä bulk- että patruunamuodossa. Tässä työssä kerrottaessa emulsioista tarkoitetaan kuitenkin lähtökohtaisesti bulk-emulsioita ja se mainitaan erikseen, jos tekstissä tarkoitetaan emulsiopatruunaa. Emulsioräjähdysaineet herkistetään panostuslaitteistolla vasta reiän panostusvaiheessa, minkä vuoksi ne ovat suhteellisen turvallisia räjähteitä. Emulsion määrää reiässä on myös helppo säädellä ja se palaa räjähtäessään täydellisesti. Räjäytysaineena bulk-emulsiot ovat saavuttaneet viime vuosina suosiota niiden panostukselle hyödyllisten ominaisuuksien ansiosta, kuten vedenkestävyys ja säädeltävyys. [15]

Emulsioiden lisäksi vaihtoehtoina ovat patrunoidut räjähdysaineet sekä anfot. Patrunoituja räjähdysaineita ovat esimerkiksi dynamiitit ja aniitti. Tähän ryhmään lasketaan mukaan myös patrunoidut emulsiot. Patrunoidut räjäytysaineet on kääritty suojaan paperin tai muovin sisälle, jolloin ne ovat helposti panostettavissa yksittäiskappaleina. Anfot valmistetaan imeyttämällä ammoniumnitraattiin polttoöljyä. Tämä työ keskittyy ainoastaan emulsioräjähteisiin, koska ne ovat käytössä kohdetyömailla tunnelinperien räjäytyksissä. Muut räjähdetyypit voivat olla käytössä toisenlaisissa räjäytyksissä, mutta niiden kappalemääräistä käyttöä on helpompi seurata eivätkä ne kuulu työn tutkimusalueeseen. [15]

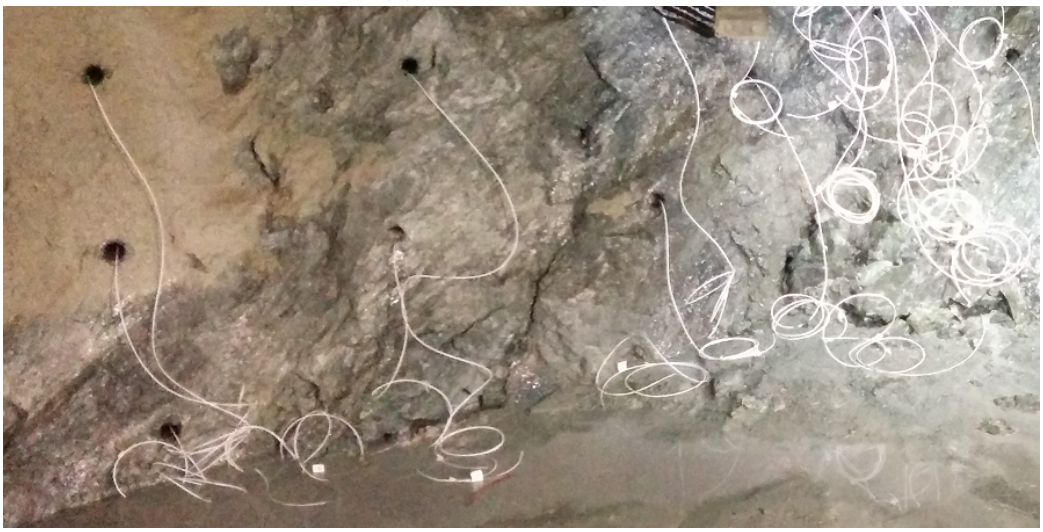
2.2.1 Emulsiolla panostaminen

Emulsioilla panostaminen tapahtuu siihen erikseen suunnitellulla panostuslaitteistolla. Laitteiston avulla kyetään säätämään reikien panostusaste sopivaksi, riippuen millaista reikää kulloinkin panostetaan. Panostuslaitteisto on asennettu panostusajoneuvoon, johon kuuluu laitteiston lisäksi myös nostettava panostuskori. Maanalaisessa panostuksessa käytetään räjähdysaineena tyypillisesti Kemiitti 810:aa, jota varten tarvitaan erityisesti sitä varten suunniteltu panostusajoneuvo. Panostusajoneuvon koko voi vaihdella käyttökohteesta riippuen ja siinä on valmius valaista pa-

nostusympäristö ja lisäksi kuljettaa työmaalle panostuksessa tarvittavat välineet ja aineet. [15]



(a) Ennen panostusta.



(b) Panostuksen jälkeen.

Kuva 2.2: Tunnelin pohja ennen ja jälkeen panostuksen

Panostuksessa emulsiomatriisi ja kaasutusliuos pumpataan letkua pitkin reikään, missä ne herkistyvät. Panostustilanteessa on paikalla usein kaksi panostajaa. Toinen panostajista käsittelee panostusajoneuvoa ja toinen syöttää panostusletkua reikiin joko panostuskorista tai maasta käsin. Reikien panostusjärjestystä ei ole ennalta määritetty. On mahdollista, että samaa reikää panostetaan useammin kuin kerran, jos ensimmäisellä panostus-

kerralla reikään ei mene tarpeeksi emulsiota. Panostuksessa reikään asetetaan myös nalli, joka räjäytysvaiheessa laukaisee panoksen. Panostetun reiän tunnistaa panostamattomasta siitä, että panostetusta reiästä tulee ohut valkoinen sytytyslanka ulos. Pohjareikien panostuksessa on huomattava, että ne eivät välttämättä ole suoraan reikiä seinässä, vaan ne voivat olla näennäisesti maan tasossa tai vedenpinnan alapuolella. Kylylahden kaivoksessa pohjareiät oli suojattu putkella, joka esti likaa ja vettä menemästä porattuun reikään. Panostuksen aikana nämä suojaputket poistettiin. Esimerkkikuva pohjarei'istä ennen ja jälkeen panostuksen on nähtävissä kuvasta 2.2.

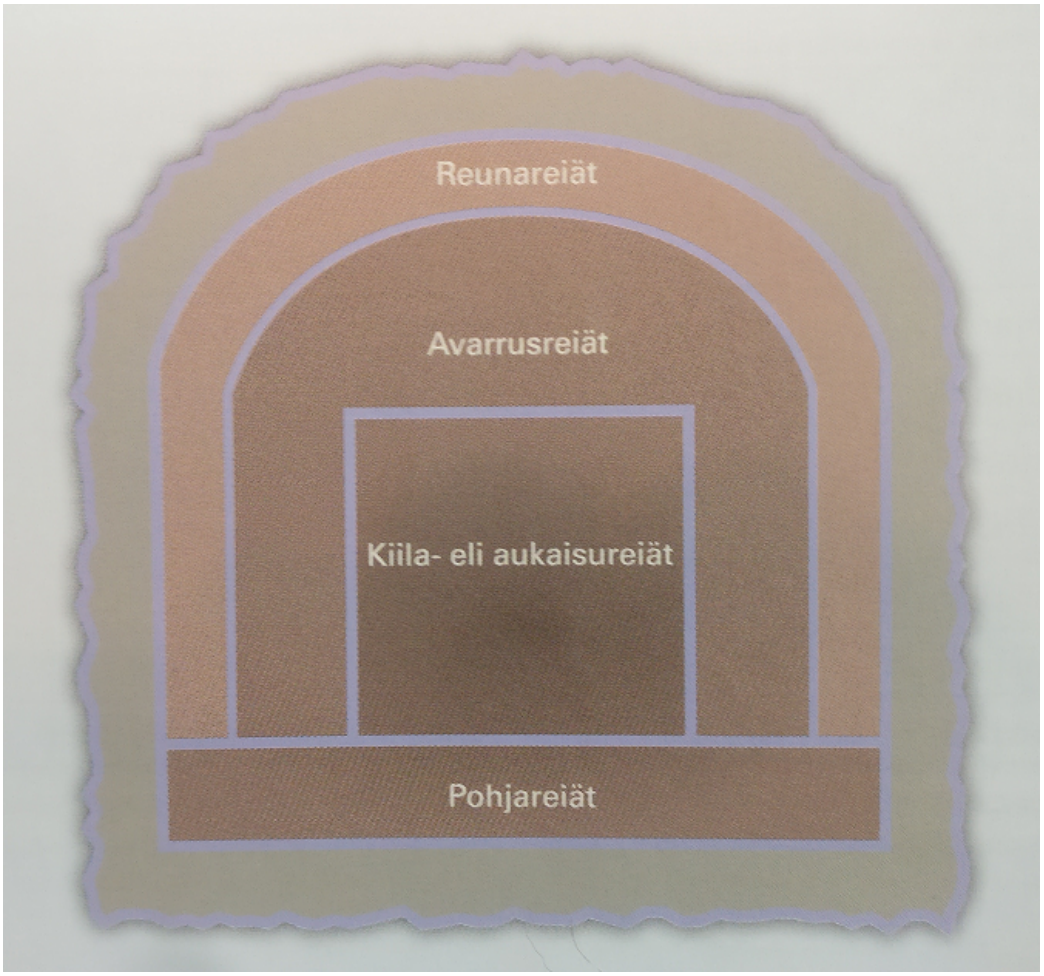
2.2.2 Räjäytyskuvio

Tunnelinperän räjäytyskuvio, tai poraus-räjäytyskaavio, tunnetaan hyvin tarkasti. Se on suunniteltu kallioon niin, että lopputuloksena on mahdollisimman tarkasti lopullista muotoa vastaava tunneliprofiili. Nykyään suunnittelu tehdään useimmin tietokoneavusteisesti. Reiät jaotellaan neljään alueeseen, jotka ovat nähtävissä kuvassa 2.3: aukaisureiät, avarrusreiät, reunareiät ja pohjareiät. Kunkin reiän läpimitta voi olla 48 mm, 51 mm tai 54 mm. Aukaisujen panostamattomat reiät jätetään kuitenkin suuremmaksi kuin 64 mm. Kunkin reiän sijainti suunnitellaan tarkasti ja reikien 3D-sijaintitiedot ovat saatavilla. Porattujen reikien sijainti vastaa hyvin suunniteltua, joten sijaintitiedot ovat tarvittaessa saatavilla myös tämän projektin käyttöön. Reikien poraustiheys vaihtelee, mutta tyypillisesti aukaisureiät ovat lähempänä toisiaan, kuin kauempana keskustaa olevat reiät. Reunareiät on kuitenkin porattu lähemmäksi toisiaan säännöllisin välimatkoin, koska niihin asetetaan pienempi ominaispanos mahdollisimman siistin räjäytysreunan saavuttamiseksi. Yksittäisten reikien välimatka toisistaan ei kuitenkaan pitäisi olla alle 10cm. [12]

2.3 Tunneliympäristö

Mikä tahansa maan alla tapahtuva toiminta aiheuttaa aina ylimääräisiä haasteita ympäristön puolesta verrattuna maanpäälliseen toimintaan. Merkittävät tekijät maan alla ovat seuraavat:

- valaistus
- kosteus
- tilanpuute
- likaisuus



Kuva 2.3: Tunnelinperän räjäytyskuvion eri reikätyypit. Aukaisureiät ovat tiheämmin kuin avarrusreiät ja niihin kuuluu myös panostamatta jätettävät suuremmat reiät. [12]

Maan alle mentäessä ensimmäisen olosuhteen muutoksen huomaa välittömästi. Jo muutaman kymmenen metrin päässä luolan suuaukosta on täysin pimeää, mikä tekee ympäristön visuaalisen tarkkailun mahdottomaksi. Pimeässä toimimiseen tarvitaan siis ulkoinen valonlähde. Kuvasta 2.4 voidaan kuitenkin nähdä, että panostuslaitteistossa on kirkkaat valaisimet, jotka valaisevat ympäristön tehokkaasti. Toisaalta voimakkaat valot aiheuttavat teräviä varjoja työskentelytilaan, mikä voi aiheuttaa hankaluuksia joillekin havainnointimenetelmille.

Toinen ympäristön rajoite on kosteus. Maan alla ilma on suhteellisen kosteaa ja maahan kertyy lätäköitä. Tämä vesi indusoituu kiviaineksesta, kos-

ka maan alle kaivautuminen muuttaa maavesien virtauksia [16]. Ympäristön kosteus itsessään ei aiheuta välttämättä suuria haasteita, mutta se tekee maasta mutaisen, ja pohjalle kertynyt vesi voi rajoittaa liikkumista ja tavaroiden ja laitteiden asettamista maahan. Kosteus aiheuttaa myös elektroniikan elinajanodotteen lyhenemistä, koska se edesauttaa laitteiston korroosiota [17].

Tämän työn kannalta oleellisin rajoite on tilanpuute. Tunneli on vain niin leveä, kuin sen käyttötarkoitukselle on välttämätöntä. On mahdollista, että panostusajoneuvon lisäksi tunneliin ei jää juurikaan tilaa ylimääräiselle laitteistolle, jolla tarkkailla panostusprosessia. Tämä vaikuttaa erityisesti näkyvyyteen panostusalueella, mikä tulee esiin seuraavissa luvuissa. Toisaalta ympäristö ei ole kahta kertaa samanlainen. Siinä missä päätylouhitut tunnelit voivat olla hyvin kapeita, ovat toiset tunnelit ja hallit hyvinkin avoimia. Tällaiset tilat louhitaan yleensä käyttämällä kattoperä-pengerlouhintaa [12]. Tällöin tunnelin yläosa louhitaan ensiksi ja pohja vasta myöhemmin, jolloin katon reunareikiä ei varsinaisesti ole. Isossa hallissa ei myöskään välttämättä ole panostettaessa seinää vieressä, johon tarkkailulaitetta voisi kiinnittää.

Viimeinen huomioitava asia tunneliympäristössä on likaisuus. Erityisesti vasta louhinnassa oleva tunneli on täynnä pölyä ja mutaa, joka väistämättä vaikuttaa joihinkin laitteistoratkaisuihin negatiivisesti. Lika voi esimerkiksi lyhentää laitteiston käyttöikää tai kameroiden tapauksessa vaatia ylimääräisiä puhdistustoimenpiteitä näkyvyyden varmistamiseksi.

Kuva 2.4: Tunnelinperä panostuksen aikana. Keskellä näkyy suuremmat panostamatta jätettävät aukaisureiät. Reikiä on tiheästi erityisesti keskellä räjäytyskuvion reunoilla.

Luku 3

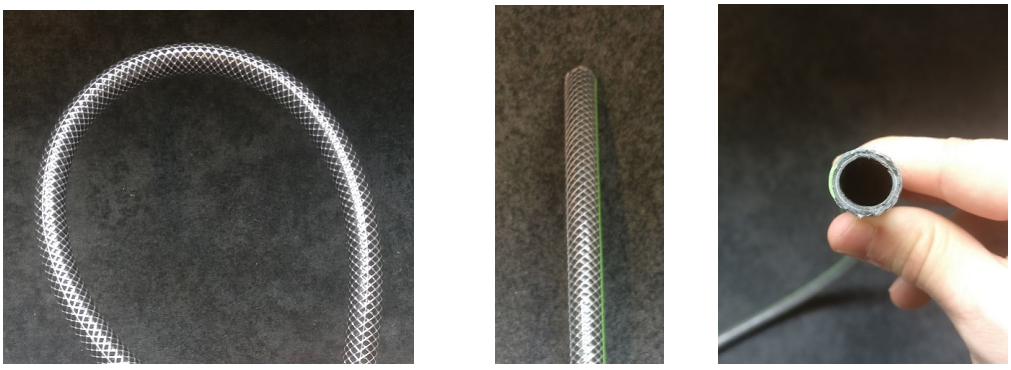
Vaihtoehtoiset menetelmät

Panostuskoneen tuottaman datan perusteella tiedetään, miten paljon emulsioräjähdettä kullakin panostuskerralla menee yksittäiseen reikään. Koska reikien panostusjärjestystä ei ole määritetty, ei kuitenkaan tiedetä, paljonko tiettyyn reikään pistettiin räjähdeainetta. On myös mahdollista, että jotakin reikää panostettiin usealla panostuskerralla, esimerkiksi jos ensimmäisellä panostuksella reikään ei tullut tarpeeksi emulsiota. Jotta saadaan kattava kuva panostusprosessista, täytyy panostusjärjestys selvittää erikseen seuraamalla panostusprosessin etenemistä. Yksinkertaisimmillaan tämä tarkoittaa, että panostaja tai tämän avustaja kirjaa reikien panostusjärjestyksen paperille tai vaihtoehtoisesti suoraan elektroniseen järjestelmään. Tämän jälkeen voidaan yhdistää tieto panostusjärjestyksestä panostuskoneen tuottamaan dataan emulsion määristä ja panostushetkistä, jolloin saadaan tietää emulsion määrä rei'issä. Panostusjärjestyksen käsin kirjaaminen ei kuitenkaan ole käytännöllinen tapa ratkaista ongelmaa. Panostusprosessi on potentiaalisesti likainen operaatio ja paperi tai kosketusnäyttö voivat mennä nopeasti käyttökelvottomiksi. Käsin kirjaamiseksi tarvitaan siis ylimääräinen henkilö pitämään panostuksesta kirjaa. Toinen vaihtoehto on hidastaa panostusprosessia niin, että yksi panostajista voi kirjata panostusjärjestyksen ylös huolehtimatta laitteiston likaantumisesta. Rakennusyhtiölle kumpikin vaihtoehto aiheuttaisi huomattavia ylimääräisiä kuluja joko lisätyövoiman muodossa tai panostusprosessin hidastumisena. Tunnelirakentaminen on aikakriittinen prosessi, sillä seuraava vaihe ei voi alkaa ennen kuin edellinen vaihe on päättynyt. Aikataulun venyminen ei ole vaihtoehto. Tämän vuoksi etsitään ratkaisua, joka ei vaadi paljoa ylimääräistä työtä panostajalta eikä aiheuta haittaa varsinaiselle panostukselle.

On tiedossa, kuinka paljon emulsiota kuluu kullakin ajanhetkellä. Jos saadaan tietää, missä panostusletkun pää sijaitsee suhteessa panostuskuvioon samalla hetkellä kuin panostetaan, saadaan myös tietää ensiksi panostetta-

va reikä ja siten myös emulsion määrä, joka kyseiseen reikään on mennyt. Edellä mainittu on tämän tutkimuksen lähtökohta ja perustamme siihen tulevan analyysin. Yksinkertaistettuna ongelman ydin on selvittää, missä letku liikkuu. Täydellisissä olosuhteissa - avoimessa ja puhtaassa tilassa - ongelma on todennäköisesti helposti ratkaistavissa. Diplomityössä kehitetyn ratkaisun tulee kuitenkin toimia tunnelissa, jossa on ahdasta ja likaista ja jossa panostajat tekevät jatkuvasti omaa työtään. Aikaisempaa tutkimusta ei ole juurikaan tehty, joten ei ole varmaa, onko ongelmaan olemassa ainuttakaan käytännöllistä ratkaisua. Tässä luvussa esitetään erilaisia menetelmiä ja arvioidaan niiden soveltuvuutta tehtävään vaadituissa olosuhteissa. Merkittävin rajoite on tilanpuute: tunnelissa on vaikeaa löytää sellaista kohtaa, josta sensori voisi havainnoida työtä luotettavasti koko panostusprosessin ajan. Esiteltyjä menetelmiä verrataan näistä lähtökohdista toisiinsa ja tämän perusteella tehdään valinta tekniikasta, jota lähdetään tutkimaan ja kehittämään eteenpäin. Alustavissa selvittelyissä tuli esiin seuraavia vaihtoehtoisia tekniikoita letkun sijainnin määrittämiseksi:

- kuva-analyysi
- pistepilvianalyysi
- lämpötilan havainnointi
- radioaaltopaikannus
- IMU



(a) Mutka (b) Suora (c) Pääty

Kuva 3.1: Letku näkyy hyvin erilaisena erilaisista kulmista

3.1 Kuvasta tunnistus

Jos ihminen suorittaisi panostuksen seurantaa, hän menisi paikan päälle, katsoisi mihin reikään panostaja laittaa räjähdysainetta ja merkitsisi sen ylös. Tunnistaminen on ihmiselle yksinkertaista, kunhan hänelle on kerrottu, mitä on tarkoitus tarkkailla. Ihminen voi käyttää tunnistamisessa hyväkseen aikaisemmin muodostettua käsitystä panostusletkun muodosta, väristä ja muista ominaisuuksista [18]. Näkyvyys ei ole myöskään ongelma, sillä ihmisellä on mahdollisuus liikkua vapaasti ympäristössä tilan sallimissa rajoitteissa. Jos panostaja hetkellisesti peittää näkyvyyden, eikä tarkkailijalla ole mahdollisuutta siirtyä, voi hän pyytää panostajaa siirtymään. Konenäkö perustuu vastaaviin periaatteisiin, mutta rajoitetummin. Koneella ei ole ihmisen tavalla vuosien varrella kertynyttä tietoa siitä, mikä on panostusletku, letku ylipäätään tai edes käsitystä ympäristön ja tarkkailtavan kohteen erosta. Koneelle nämä kaikki asiat täytyy joko opettaa tai vaihtoehtoisesti antaa joukko sääntöjä, joilla se voi löytää halutun piirteen kuvasta. Koneella ei ole myöskään lähtökohtaisesti mahdollisuutta liikkua parempaan asemaan näkyvyyden peittyessä. Tämä ei tarkoita, ettemmekö voisi hyödyntää konenäköä, mutta se vaatii ongelmien kiertämistä.

Kuvasta tunnistamisella tarkoitetaan tässä yhteydessä konenäön käyttämistä letkun tunnistamiseen valokuvasta. Yksinkertaisimmillaan se tarkoittaa letkunväristen ja kokoisten alueiden löytämistä kuvasta [19] [20]. Käytännössä näin yksinkertaiset menetelmät eivät voi luotettavasti toimia ongelman ratkaisemisessa. Ne vaativat toimiakseen hyvin rajatun ympäristön, jossa kuvausalueella ei ole tarkkailtavaan kohteeseen sekoittuvia elementtejä. Panostusprosessi ei ole tällainen ympäristö. Ei voida taata, että ympäristössä ei olisi muita letkun värisiä kohteita. Letku ei ole myöskään tarkasti määritetty kohde, vaan sen muoto vaihtelee huomattavasti riippuen katselukulmasta. Esimerkki kameran näkemästä letkusta erilaisista kuvakulmissa on nähtävissä kuvasta 3.1. Yhdestä kulmasta letku näyttää kaarelta ja toisesta suorakaiteelta. Tämä tarkoittaa, että voidaan hylätä letkun tunnistamisen muodosta.

Edellämainittuja ongelmia voi yksinkertaistaa varmistamalla, että letkussa on helposti tunnistettava piirre, joka varmasti erottuu ympäristöstä. Yksi vaihtoehto tähän on letkun päähän sijoitettava kirkkaanvärinen suppilo tai pyöreä levy. Tunneliympäristön värimaailma koostuu pääasiassa harmaista sävyistä, joten riittää valita väri, joka ei esiinny työmiesten vaatetuksessa tai panostuslaitteistossa. Kuvan 3.2 perusteella nähdään, että työmiesten vaatteissa esiintyy pääasiassa sinistä ja keltaista, kun taas panostuskori on oranssin värinen. Erottuvimmat värit olisivat siis todennäköisesti joko punaisen

Kuva 3.2: Töölön parkkihallin panostusta. Sivuilla on paljon tilaa, mutta maassa näkyy lätäköitä ja seinämä on hyvin epätasainen.

tai vihreän sävyjä. On kuitenkin vaikea taata, että valittua väriä ei esiintyisi koskaan ympäristössä. Tähän vaikuttaa varsinkin tunnelin valaisuolosuhteet. Panostustilanne on aina hyvin valaistu, mutta toisaalta voimakkaat valot aiheuttavat teräviä varjoja tunnelin seinämään. Algoritmin voi olla vaikea tunnistaa kohdetta samaksi silloin, kun se sijaitsee varjossa ja silloin kun kohde on kirkkaimmassa valokeilassa [21]. Huomionarvoinen asia ylimääräisten kappaleiden käyttämisessä tunnistamisen apuvälineenä on, että ylimääräisellä osalla letkun päässä voi olla työtä vaikeuttavia vaikutuksia. Koska panostusletku työnnetään panostettavaan reikään, täytyy kappaleen olla liikuteltavissa letkua pitkin. On vaikea sanoa varmasti, kuinka paljon se häiritsee panostajan työtä. Voidaan kuitenkin kuvitella tilanne, jossa letku vedetään ulos reiästä ja suppilo irtoaa letkusta.

Kohteen seurannan ei tietenkään tarvitse perustua ainoastaan väriin ja muotoon. Kaksi muuta hyödynnettävää tekniikkaa ovat tekstuurin ja liikkeen seuranta. Tekstuuri helpottaa huomattavasti kappaleen seuraamista huolimatta erilaisten valaisuolosuhteiden aiheuttamista värimuunnoksista. Liikkeen hyödyntäminen puolestaan tulee luonnollisesti, jos saatavilla on videokuvaa tai ainakin kuvia, jotka on otettu lyhyen ajan sisällä toisistaan. Tällöin voidaan käyttää hyväksi oletusta, että tarkkailtu kohde ei ole juurikaan siirtynyt edellisestä sijainnistaan. Liikkuvaa kohdetta havaitessa taas voidaan

arvioida kohteen nopeutta ja estimoida letkun tulevaa sijaintia etukäteen. [22]

Mihin kuvan luokittelu osiin, joista letku löytyy ja toisiin osiin, joissa ei ole letkua sitten perustuu? Vastaus löytyy kuvasta eristetyistä piirteistä ja niiden pohjalta luoduista algoritmeista. Luokitteluprosessi voidaan määritellä seuraavasti: ensimmäiseksi kuvalle suoritetaan esiprosessointi, jossa kuvan kontrastia ja kirkkautta säädetään sopivaan tasoon. Tarkka säätö riippuu kuitenkin käyttökohteesta, eikä sitä voida etukäteen tietää. Toisessa vaiheessa kuvasta eristetään piirteet. Tämä tapahtuu esimerkiksi suorittamalla reunanhavainnointialgoritmi kuvalle, jolloin tuloksena on tarkkarajaiset reunat kuvan värirajoista. Yksi yleisesti käytetty tapa on käyttää niin kutsuttuja Haar-piirteitä [23]. Haar-piirteet voivat olla esimerkiksi reunoja tai viivoja. Algoritmi etsii kuvasta siihen parhaiten sopivat piirteet, joiden perusteella tarkkailtava alue joko todetaan potentiaalisesti kuuluvaksi etsittävään kappaleeseen tai ei-kuuluvaksi. Viimeinen vaihe luokittelussa on algoritmin opettaminen tiettyyn luokittelutapaukseen. Algoritmin opettaminen tapahtuu näyttämällä sille tuhansia kuvia tunnistettavasta asiasta, mikä tämän projektin tapauksessa on panostusletku. Algoritmi etsii prosessoiduista piirteistä sellaiset, jotka ovat olemassa suurimmassa osassa oikeaksi todetuista opetuskuvista. Suorittamalla saman piirteiden haun prosessoitavalle kuvalle, voidaan kuva luokitella opetetulla algoritmilla. [24]

Viime vuosina kappaleiden seurannassa ja tunnistamisessa ovat yleistyneet syväoppiminen ja neuroverkot [25]. Neuroverkot toimivat pääpiirteittäin hyvin samalla tavalla, kuin edellä mainittu piirteiden haku ja opettaminen. Ne eroavat kuitenkin siinä, että piirteitä ei haeta erikseen, vaan neuroverkko muodostaa esimerkkien avulla itsenäisesti käsityksen siitä, miltä kappale näyttää. Niille ei siis määritellä välttämättä valmiiksi sääntöjä. Tässä projektissa ei käsitellä tällaisia koneoppimismenetelmiä tarkemmin, koska ne vaativat huomattavan määrän esimerkkidataa neuroverkon rakentamiseksi [26]. On kuitenkin hyvä tiedostaa niiden käyttökelpoisuus yleisesti kuvien luokittelussa.

2D-konenäön periaatteellinen ongelma liittyy maailman kolmiulotteisuuteen. Käsitellyt valokuvaan perustuvat menetelmät keskittyvät lähinnä kappaleiden tunnistamiseen tapauksissa, joissa kamera pysyy paikallaan ja ollaan kiinnostuneita ainoastaan tarkkailemamme kohteen sijainnista valokuvassa. Tämä ei ole riittävä tarkkuus tutkimusongelman ratkaisemiseksi. Muistetaan, että panostusletkun sijainti täytyy pystyä yhdistämään räjäytyskuvion koordinaatistoon. 2D-sijainti kamerakoordinaatistossa ei siis riitä, vaan kuvasta täytyy myös pystyä päättelemään kohteen etäisyys kamerasta. Tunnistetun kohteen sijainti täytyy olla mahdollista muuttaa kamerakoordinaatistosta räjäytyskuvion koordinaatistoon. Todennäköisesti riittää, että kamera-

koordinaatisto liitetään panostuskoordinaatistoon käsin myöhemmin, joten tätä osaa ongelmasta ei lähdetä ratkaisemaan tarkemmin. Sen sijaan 2D-kuvasta saatavan tiedon muuttaminen 3D-dataksi on monimutkaisempi operaatio. Tämä ei onnistu yhden tavallisen kameran tuottaman kuvan avulla, vaan siihen tarvitaan joko vähintään kaksi kameraa [27] tai yksi liikkuva kamera [28][29]. Lisäksi on olemassa erityisesti etäisyystarkkailuun luotuja kameroita. Niitä käsitellään tarkemmin kappaleessa 3.2. Kahta kameraa käyttävää systeemiä voidaan kutsua stereokameraksi.

Kaikkien kameroihin perustuvien ratkaisuiden ongelmana on kameran sijoittaminen. Letkun löytäminen kuvasta on mahdotonta, jos se ei ole edes näkyvissä. Tämä on tilanne, jos panostaja kulkee kameran ja panostettavan reiän välissä. On teoriassa mahdollista käyttää useampaa kameraa katveiden kattamiseksi, mutta tunneliympäristössä voi olla vaikea löytää sellaisia paikkoja, mihin useamman kameran voisi asentaa.

3.2 Muodon tunnistaminen pistepilvestä

Aikaisemmin käsiteltiin kuvasta tunnistusta mahdollisena menetelmänä letkun sijainnin määrittämiseksi. Herää kysymys: jos joka tapauksessa tiedot tarvitaan 3D-koordinaatistossa, eikö olisi viisaampaa käyttää suoraan 3D-kameraa tähän tarkoitukseen. 3D-kamera tarkoittaa tässä yhteydessä kameraa, joka osaa tuottaa pistepilven kuvattavasta kohteesta jollakin menetelmällä. Toinen laitteesta käytetty nimi on syvyysskanneri. Yksinkertaisimmillaan 3D-kamera voidaan rakentaa kahden tavallisen kameran avulla [27], mutta käytännöllisempää on käyttää erikseen tätä varten suunniteltua laitteistoa. Näin vältytään ylimääräiseltä kuvien prosessoinnilta. Tällaiset laitteet perustuvat esimerkiksi laser-keilaukseen, ultraääneen tai infrapunavaloon.

Kuten kuvasta tunnistamisessa, myös pistepilvimenetelmissä on voimassa samoja rajoitteita. Kameran sijoittaminen on edelleen kysymysmerkki, kuten on myös kameran paikannus referenssimalliin nähden. Eroavaisuus syntyy siitä, että 3D-pisteitä ei käytetä ainoastaan loppudatassa, vaan myös letkun tunnistamisessa. Etuna tässä on, että letkuun ei välttämättä tarvita ylimääräisiä kappaleita, eli panostajan työ vaikeutuu niin vähän kuin mahdollista. Tunnistamista helpottavien osien puuttuminen aiheuttaa toisaalta myös ylimääräisiä haasteita. Jos letkun reiässä oleva pää ei näy, mutta joku muu osa näkyy, saadaan virheellisiä tuloksia letkun pään sijainnista. Toinen potentiaalinen ongelma on muunlaiset virhetulokset. Ympäristössä voi olla muita kappaleita, jotka ovat letkunmuotoisia ja näistä kappaleistä pitää pystyä päättelemään, mikä on todellinen letku ja mikä väärä.

Jos jätetään pois mahdollisuus käyttää syväoppimista letkun tunnistamiseen, voidaan letku tunnistaa pistepilvestä etsimällä letkun osia sylinterin matemaattisen mallin perusteella [30]. Letkun mallia ei voida suoraan etsiä, sillä se ei pysy samassa muodossa käytön aikana. Kokonainen letku voidaan paikantaa löytämällä lähekkäiset sylinterit. Tämänkin jälkeen on vielä ongelmia ratkaistavana. Lähtökohtaisesti ei ole tiedossa, kumpi pää letkua on lähempänä reikää. Letku voi näkyä myös useassa osassa, jolloin algoritmin pitäisi pystyä päätellä, mikä letkun osista on relevantti tarkkailun kannalta. Luotettavuus on myös kyseenalaista: kamera ei voi nähdä letkua kokonaan. Letku täytyy pystyä tunnistamaan myös silloin, kun siitä näkyy vain toinen puoli. Kameran resoluutio ja mittaustarkkuus vaikuttavat myöskin tulokseen.

3.3 Lämpötilasensori

Lämpötilasensoria, eli käytännössä infrapunakameraa, esitettiin yhtenä vaihtoehtona letkun tunnistamiseksi muusta ympäristöstä. Idea perustuu ajatukselle, että emulsion - ja siten myös letkun - lämpötila eroaisi ympäristöstä. Selvityksen perusteella kävi kuitenkin ilmi, että räjähdysaineen lämpötila ei merkittävästi eroa ympäristön lämpötilasta. Emulsio saattaa olla lämpimämpi, kun se tuodaan tunneliin, mutta lämpötila laskee hiljalleen ympäristön lämpötilaan, joka säilyy lähes vakiona ympäri vuoden [31]. Panostaja itse on merkittävämpi lämmönlähde kuin panostusletku.

3.4 Radioaaltojen hyväksikäyttö paikannuksessa

Radioaaltojen käyttöä letkun paikannuksessa voidaan verrata GPS:ään, mutta paikallisessa ympäristössä. Menetelmässä letkun päässä sijaitsevasta radiolähettimestä lähetetään signaali, jonka vähintään kolme tunnetuissa paikoissa sijaitsevaa vastaanotinta havaitsevat. Signaalin tulosuunnista voidaan kolmiomittauksella laskea tarkka paikka, josta signaali on lähtenyt. Mitä useampi sensori havaitsee signaalin, sitä tarkempi arvio voidaan muodostaa lähtöpisteestä. [32]

Kolmiomittauksen varjopuolena on sen epätarkkuus. Se on altis monitieetenemiselle, jossa sama signaali kulkee useita reittejä esimerkiksi heijastusten kautta [33]. Signaalivastaanottimia tarvittaisiin huomattava määrä sopivan tarkkuuden saavuttamiseksi. Toinen ongelma on laitteistovaatimukset. Tarkan suunnan määrittämiseksi vastaanottimessa täytyy olla useita antenneja [32]. Muitakin tapoja päätellä signaalin tulosuunta on olemassa. Yksi

tapa on mitata signaalin voimakkuutta, mutta tämä kärsii huomattavasti suuremmasta mittausepätarkkuudesta [34]. Toinen tapa on synkronoida vastaanottimien kellot, jolloin signaalin vastaanottohetkistä voidaan laskea tulosuunta [35]. Tässäkin tapauksessa mittausepätarkkuus on suuri ja lisäksi kellojen synkronointi täytyisi tehdä nanosekuntitarkkuudella.

Mahdollisesti ylitsepääsemätön ongelma, joka koskee radioaaltojen hyväksikäyttöä liittyy vastaanottimien asetteluun. Jotta niitä voi käyttää paikannukseen, vastaanottimien sijainnit täytyy tuntea tarkasti. Panostuskori ei sovellu tällaisten vastaanottimien sijoituspaikaksi, koska sen sijaintia suhteessa räjäytyskuvioon ei tunneta ja se liikkuu jatkuvasti. Manuaalinen räjäytyskuvioon kalibrointi taas on hankalaa ilman visuaalista vertailukohtaa. Vastaanottimien sijainti täytyisi etukäteen määrittää suhteessa räjäytyskuvioon ja niiden paikat mitata tarkasti ennen panostuksen aloittamista. Tämän työn lisäksi panostajan täytyisi pitää huolta lähettimestä, että se pysyy letkussa kiinni.

3.5 Inertial Measurement Unit, IMU

Inertial Measurement Unit, eli IMU, on liikkeen seuraamiseen tarkoitettu laite, joka perustuu gyroskooppiin ja kiihtyvyysanturiin. Sen avulla voidaan laskea missä asennossa laite on mihin suuntaan se liikkuu. IMU:en ongelma on yleisesti niiden kasvava virhe. Koska se ei havaitse paikkaa suoraan, vaan kiihtyvyyden kautta, pienikin virhe kasvaa lopulta merkittävän isoksi ja laskettu sijainti vääräksi. [36]

Teoriassa tarpeeksi hyvän IMU:n kanssa tätä ongelmaa voitaisiin kiertää lukitsemalla sijainti lähimpään reikään silloin kun panostusta suoritetaan. Tällöin virhe ei ehdi kasvaa liian suureksi ja tulokset pysyvät tarkkoina. Käytännössä ei voida kuitenkaan olettaa, että panostajat siirtäisivät panostusletkun suoraan reiästä toiseen: panostusletku voi tyypillisessä käytössä hetkellisesti pudota, jolloin tärähdys sekoittaa sijaintilaskut täydellisesti ja IMU pitäisi kalibroida uudestaan. Kalibrointi on IMU:n tapauksessa sen hyvä puoli. Tietty reikä, esimerkiksi keskimmäisen aukaisureiän, voidaan määrittää kalibrointireiäksi. Tällöin ensimmäistä reikää panostettaessa tiedetään sijainti jo valmiiksi suhteessa panostuskuvioon.

Toinen ongelma IMU:en käytössä liittyy tiedonkulkuun ja virtaan. Panostusletkua ei voi johdottaa: se on käytännössä mahdotonta letkun liikkumisen vuoksi. Laitteen täytyy siis toimia itsenäisesti. Itsenäinen toiminta vaatii ainakin akun ja tietoliikenneyhteydet, jolloin laitteesta voi tulla kookas. Kokoon vaikuttaa myös käytetty IMU, sillä pienemmät IMU:t tapaa-

vat olla vähemmän tarkkoja [37]. Laitteen täytyisi olla myös hyvin suojattu kestämään kulutusta ja emulsion hapettavuutta [38].

3.6 Menetelmän valinta

Ennen kattavaa testaamista on vaikea sanoa mikä menetelmistä on paras ratkaisemaan ongelman ratkaisemiseksi. Ei ole myöskään takeita, että mitään menetelmistä olisi edes mahdollista toteuttaa niin, että se toimii luotettavasti. Kamerapohjaisisssa menetelmissä on ongelmana näkyvyyden varmistaminen, mutta hyvissä olosuhteissa niiden pitäisi olla suhteellisen varmoja toiminnassaan. Kuva-analyysiä ja pistepilveä verratessa kuva-analyysi on teoriassa yksinkertaisempi toteuttaa, mutta näiden välillä valitaan pistepilvien hyödyntäminen helpomman syvyyslaskennan takia.

Lämpötilasensori hylätään vaihtoehtona teknisesti toteutuskelvottomana. Radioaaltojan hyväksikäyttö on ehkä luotettavin menetelmä kunhan sensoreita on tarpeeksi. Käytännössä se on kuitenkin aivan liian työläs sensorien sijoittamiseen kuluvasta ajasta johtuen. Jos panostuskorin sijainti tunnettaisiin ja sensorit voitaisiin sijoittaa siihen, olisi radiopaikannus erittäin hyvä vaihtoehto huolimatta ylimääräisen kappaleen panostusletkun käsittelyyn aiheuttamasta lisävaivasta.

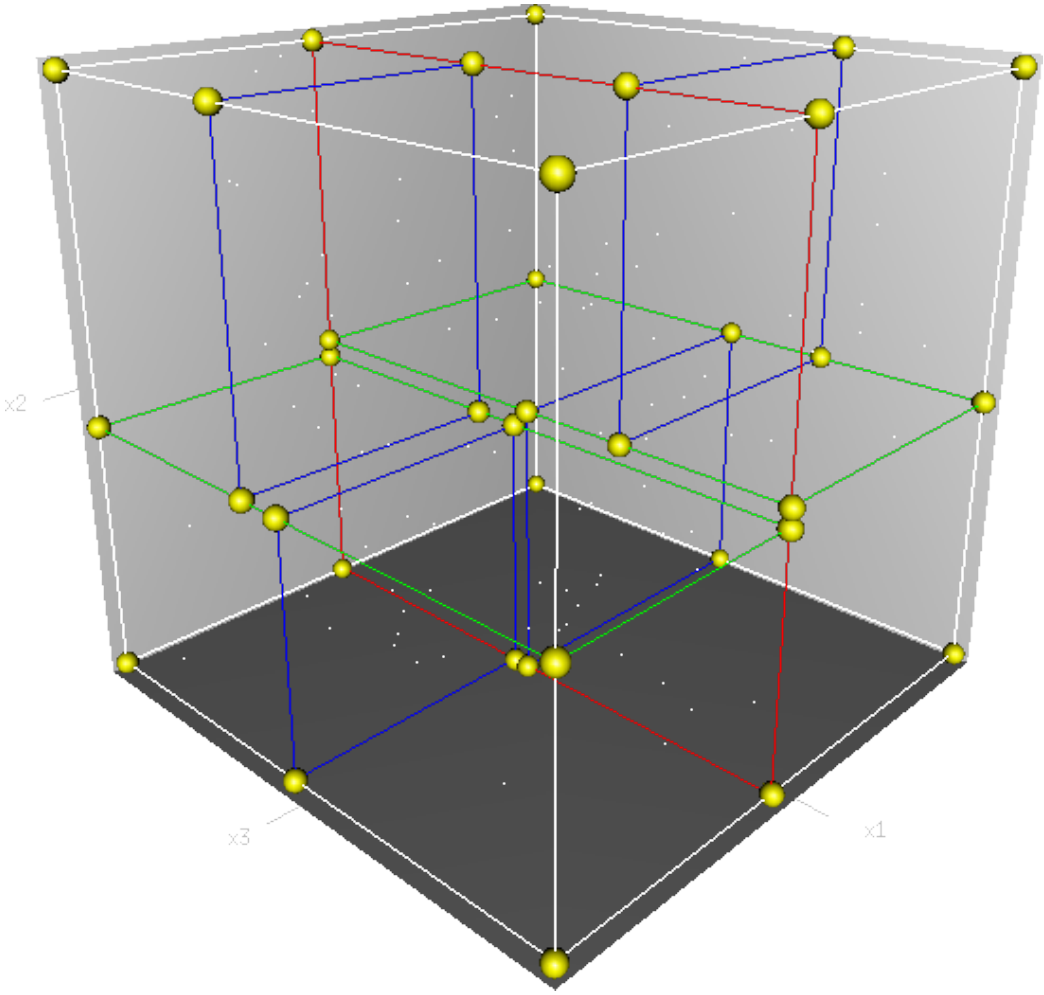
IMU soveltuisi tehtävään optimiolosuhteissa hyvin huolimatta kertyvästä virheestä. Käytännössä kuitenkin yksikin virhe, kuten pudonnut letku, tuhoaa mittaustulokset. Verratessa IMU:a ja pistepilvimenetelmiä potentiaalisina vaihtoehtoina päätettiin menetelmäksi lopulta valita *muodon tunnistamimen pistepilvestä*, koska se ei vaadi ylimääräisiä osia panostusletkuun.

Luku 4

Pistepilvien teoriaa

Reaalimaailma kuvataan tietokonegrafiikassa yleensä tasoina, jotka määrittyvät kolmen avaruuden pisteen perusteella. Pistepilvi on vastaava kuvaus maailmasta, mutta pisteet eivät lähtökohtaisesti tiedä suhdettaan muihin pisteisiin. Ne siis muodostuvat ainoastaan pisteistä, eivät esimerkiksi kappaleista tai edes pisteiden välisistä viivoista. 3D-kamerat tuottavat tällaisia malleja ympäristöstään. Ihminen voi näytölle piirretystä pistepilvestä helposti nähdä muotoja ja ymmärtää, mitä kuva on esittänyt. Tietokoneelle tällainen tunnistaminen ei ole helppoa: yksinkertainenkin pistejoukon tulkinta, esimerkiksi tasojen määrittäminen läheisten pisteiden perusteella, vaatii oman algoritminsa. Yhdistämällä pistepilvialgoritmeja toisiinsa tietokoneohjelman ymmärrys ympäristöstään kasvaa ja se voi alkaa esittää arvioita pisteiden muodostamista tasoista tai kappaleista.

Panostusprosessissa käytettävän letkun tunnistaminen pistepilvimenetelmien avulla vaatii useiden eri pistepilviin soveltuvien algoritmien hyväksikäyttämistä. Vaikka algoritmit tulisivat valmiina suoraan kirjastosta, on olennaista ymmärtää, mitä algoritmin taustalla tapahtuu ja miksi juuri tätä algoritmia käytetään eikä jotakin toista. Joidenkin algoritmien kohdalla ymmärrys auttaa myös niiden virittämisessä, jotta ne löytävät datamassasta juuri ne piirteet joista ollaan kiinnostuneita. Tämä luku selittää prototyypissä käytettävien pistepilvimenetelmien periaatteet näistä näkökulmista ja valottaa tarpeellisella tasolla niiden teoreettista taustaa. On huomioitava, että yksikään algoritmeista itsessään ei ole taika-avain, joka ratkaisee koko ongelman. Ainoastaan niiden yhteispelillä voidaan löytää niinkin näennäisen yksinkertainen muoto kuin letku. Tulkintaa vaikeuttaa esimerkiksi se, että jos algoritmin halutaan löytävän sylinteri, se voi palauttaa minkä tahansa sylinterin - ison tai pienen. On käyttäjän käsissä, hyväksyykö hän koneen antaman tuloksen. Algoritmien käyttöä ja niiden vuorovaikutusta on kuvattu tarkemmin luvussa 5 Prototyypin toteutus.



Kuva 4.1: Kolmiulotteiden kd-puu. Valkoiset pisteet muodostavat jaettavan pistepilven. Punainen viiva merkitsee jakoa ensimmäisen ulottuvuuden mukaan, vihreät viivat toisen ulottuvuuden mukaan ja siniset viivat kolmannen ulottuvuuden mukaan. [39]

4.1 Kd-puu

Kd-puu (engl. *k-d tree*), on yleisesti käytetty datarakenne, joka organisoi pisteet k-ulotteisessa avaruudessa. Se on binäärinen etsintäpuu, jota käytetään esimerkiksi etäisyys- ja lähimmän naapurin hakuihin. Puuta muodostettaessa jokainen uusi piste sitoutuu yhteen k:sta ulottuvuudesta. Tyypillinen tapa muodostaa kd-puu menee seuraavasti. Ensimmäisenä valitaan akseli, jonka mukaan jäljellä olevat pisteet jaetaan: kolmiulotteisessa avaruudessa x, y tai z-akseli. Akselin valinta on kiertävä, eli jokainen akseli toimii vuorotellen ja-

kavana ulottuvuutena. Kun kaikki ulottuvuudet on käyty läpi, aloitetaan uudestaan ensimmäisestä ulottuvuudesta. Jakamattomista pisteistä lasketaan edellä valitun akselin mukainen mediaanipiste, joka toimii ankkurina operaatiolle. Kaikki loput pisteet jaetaan vasemmalle tai oikealle puolelle mediaanipisteestä. Sen jälkeen operaatio suoritetaan molemmille oksille erikseen kierrossa seuraavana olevassa ulottuvuudessa. Tätä jatketaan kunnes kaikki avaruuden pisteet on sijoitettu omaan soluunsa, johon ei kuulu muita pisteitä. [40]

Kd-puu on erittäin käytännöllinen työkalu pistepilvien käsittelyssä. Ilman tehokasta hakurakennetta, olisivat useat käytetyt algoritmit erittäin hitaita. Tämä johtuu siitä, että kd-puu jakaa etsittävän alueen pienempiin osiin, jolloin vertailuoperaatioita voidaan suorittaa kokonaisille soluille yksittäisten pisteiden sijaan. Esimerkki kolmiulotteisen kd-puun muodostuksesta ja soluihin jaosta on nähtävillä kuvassa 4.1. Kd-puun lisäksi toinen yleisesti käytetty binäärinen hakupuu on octree. Octree on lähes kuin kd-puu, mutta se jakaa solut aina kuution muotoisiksi. Kd-puu on kuitenkin tehokkaampi yleisessä tapauksessa. [41]

4.2 Näytteiden harventaminen

Uudenaikaisten 3D-kameroiden ottamissa kuvissa voi olla satojatuhansia tai jopa miljoonia pisteitä. Jotkut operaatiot hidastuvat nopeasti pisteiden määrän kasvaessa yli tietyn tason. Tämän projektin tapauksessa tällainen epälineaarisesti hidastuva algoritmi on pistepilvien erottelu. Näytteiden harventaminen (engl. *downsampling*) vastaa tähän ongelmaan pienentämällä laskuihin käytettävien pisteiden määrän sopivaksi kyseiselle operaatiolle [42]. Harventamisen kanssa täytyy olla varovainen, sillä jos pilveä harvennetaan liikaa, alkaa harvennus vaikuttaa merkittävästi algoritmin lopputulokseen. Esimerkiksi pisteet jotka muodostavat pallon voivat harvennuksen jälkeen muodostaa enää kuution.

Tässä työssä näytteiden harventamiseen käytetään niin kutsuttua vokseliverkkoa (engl. *voxel grid*). Vokseliverkolla suodattaessa voidaan ajatella, että koko pisteavaruus jaetaan samankokoisiin laatikoihin, eli vokseleihin. Jokaiselle laatikolle lasketaan sen sisällä olevien pisteiden painopiste, joka on suodatetussa pilvessä ainoa kyseisen vokselin sisällä oleva piste. Näin saatu pistepilvi on niin harva kuin mikä on määritelty vokselin kooksi. Keskimäärin lähimpien pisteiden etäisyys on siis suunnilleen yhtä suuri kuin vokselin särmien pituus. [43]

4.3 Normaalien estimointi

Useita pistepilvialgoritmeja varten on tarpeellista tietää, millaiset normaalit pisteillä on. Tavallisesta 3D-mallista tämä on helppoa, koska jokaiseen pisteeseen liittyvät naapuripisteet tunnetaan. Pistepilvien tapauksessa ei ole olemassa valmiina tietoa pisteiden välisistä suhteista. Tämän vuoksi normaalit täytyy arvioida erikseen. Erilaisia normaalinestimointimenetelmiä on useita, mutta tähän projektiin valittiin yksinkertaisen menetelmä, jota PCL käyttää.

Algoritmissa pisteen normaalia approksimoidaan sen läheisimpien pisteiden muodostaman tason avulla. Käytännössä tämä tapahtuu naapuruston kovarianssimatriisin ominaisvektori- ja ominaisarvoanalyysillä. Normaalin suunta määritetään siten, että normaalit kohdistuvat kameraan päin. Normaalien estimoinnissa voidaan parametrina määrittää säde, joka kertoo normaalin laskemisessa käytettyjen naapureiden määrän. [44]

4.4 Pistepilvien erottelu

Pistepilviä käsiteltäessä voi olla käytännöllistä löytää eroavaisuudet kahden pistepilven väliltä. Joko halutaan selvittää, mikä kuvassa on muuttunut tai vaihtoehtoisesti halutaan vain rajata jokin operaatio pienemmälle alueelle. Tyypillisessä esimerkkitapauksessa pilvestä halutaan erottaa tunnettu tausta. Kahdesta pistepilvestä ei voida kuitenkaan suoraan laskea niiden erotusta, sillä tällaista operaatiota ei ole määritelty. Pilvissä voi olla esimerkiksi eri määrä pisteitä tai niiden järjestys datarakenteessa voi vaihdella. Jos pistepilvi on tarkasti määritelty vaikkapa niin, että pisteet on jaoteltu tasaisesti, on mahdollista tehdä yksinkertainen haku ja tarkistaa, löytyykö testattava piste myös toisesta pilvestä. Tilanne on toinen, kun sama fyysisen maailman piste saa kahdella kuvauskerralla erilaisen sijainnin mittausvirheestä johtuen. Niin on tilanne tämän projektin kohdalla, kun käytössä on 3D-kamera pistepilven tuottamista varten.

Ongelman ratkaisemiseksi lisätään erotusalgoritmiin hakuehto, joka määrittää kuinka kaukana toisistaan olevat pisteet tulkitaan samoiksi. Algoritmissa määritellään pääasiallinen pilvi sekä vertailupilvi. Lopputulokseen valitaan pääasiallisesta pilvestä vain ne pisteet, jotka eivät ole lähellä yhtäkään toisen pilven pistettä. Tämä tapahtuu hyväksikäyttäen Kd-puuta. Jokaiselle pääasiallisen pilven pisteelle suoritetaan lähimmän naapurin haku. Jos pisteen lähin naapuri on kauempana kuin asetettu kynnysarvo, otetaan piste mukaan erotuspilveen.

4.5 RANSAC

Muotojen tunnistamiseen 3D-datasta on olemassa kaksi yleisesti käytettyä menetelmää: Houghin algoritmi, sekä RANSAC [45][46]. Houghin algoritmi on pääasiassa käytössä tasojen tunnistamiseen, johtuen sen suuresta muistintarpeesta sekä prosessointinopeudesta [47]. Vuonna 2005 Rabbanin ryhmä kuitenkin esitti Houghin algoritmia käyttävän menetelmän sylintereiden tunnistamiseksi, joten sekään ei ole täysin pois suljettu menetelmä mahdollista tulevaa kehitystä ajatellen [48]. Tässä projektissa tuotetaan dataa kuitenkin paljon, joten ei ole mahdollista käyttää liikaa aikaa sylintereiden tunnistamiseen. Sen vuoksi tähän työhön valitaan RANSAC pääasialliseksi menetelmäksi.

RANSAC on lyhenne sanoista Random Sample Consensus, eli Satunnaisotannan Konsensus. Menetelmä on ollut yleisesti tunnettu jo 80-luvulta lähtien, kun Fischler ja Bolles esittelivät sen artikkelissaan. Vaikka pistepilvien tapauksessa RANSAC:ia käytetään muotojen tunnistamiseen, on se yleisempi menetelmä mallin sovittamiseen kokeelliseen dataan. Yksinkertaisimmillaan tällaiset menetelmät, kuten pienimmän neliösumman menetelmä, käyttävät hyväkseen kaikkea dataa ja yrittävät sovittaa mallin vastaamaan koko datajoukkoa niin täydellisesti kuin mahdollista. Tällaiset menetelmät epäonnistuvat, jos datapisteet eivät ole tarpeeksi lähellä algoritmin tarjoamaa mallia. Tämä johtuu siitä, että algoritmeihin ei ole sisäänrakennettua mekanismia virheiden havaitsemiseksi. Sen sijaan ne tekevät niin kutsutun sileysoletuksen ja keskiarvoistavat dataa, jolloin pienet epätarkkuudet siloittuvat ja data saadaan sopimaan malliin. Ne eivät siis sovellu tilanteeseen, jossa lähtödatassa on paljon vihreitä tai se sisältää kohinaa. RANSAC vastaa tähän ongelmaan päinvastaisella tavalla. Sen sijaan, että yritettäisiin parametrisoida malli vastaamaan koko pistejoukkoa, ottaa RANSAC mahdollisimman pienen osan kaikesta datasta ja sovittaa mallin sitä vasten. Sen jälkeen RANSAC kasvattaa sovittamiseen käytettyä pistejoukkoa niin, että se edelleen sopii malliin tarpeeksi hyvin. [46]

PCL:n RANSAC-toteutus vaatii toimiakseen kolme parametria: virhetoleranssin, iteraatioiden määrän, sekä onnistumistodennäköisyyden. Virhetoleranssi kertoo, kuinka kaukana estimoidusta mallista pisteet voivat sijaita niin, että ne vielä lasketaan kuuluvaksi malliin. Iteraatioiden määrä kertoo, kuinka monta kertaa algoritmi suoritetaan enimmillään, mikä vastaa alkuperäisessä tutkimuksessa kokeiltavien pistejoukkojen määrää. Onnistumistodennäköisyys puolestaan kertoo, kuinka todennäköisesti mallin täytyy olla paras löydettävissä oleva, että algoritmin suorittaminen lopetetaan. Viimeinen parametri pitää sisällään RANSAC:n ymmärtämisen

kannalta tärkeän havainnon. RANSAC ei ole deterministinen algoritmi, vaan samoilla parametreilla voi tulla täysin erilainen tulos. Esimerkiksi todennäköisyysparametrilla 0,95 yksi kahdestakymmenestä algoritmin suorituksesta palauttaa epäoptimaalisen tuloksen. Parametrien valinnassa täytyy tiedostaa, ettei ole olemassa taika-arvoja, jotka saavat algoritmin sopimaan minkä tahansa mallin löytämiseen mistä vain datasta, vaan datan laatu ja ominaisuudet vaikuttavat aina parametrien valintaan. [49]

Kun RANSAC:a käytetään muodon tunnistamiseksi pistepilvestä, on prosessi seuraavanlainen. Oletetaan pistejoukon ominaisuudet tunnetuksi, jotta sille on pystytty arvioimaan tarpeeksi hyvät parametrit algoritmille. RANSAC ei pysty automaattisesti tunnistamaan mitä tahansa muotoa, joten sille täytyy myös antaa matemaattinen malli, jonka parametreja yritetään optimoida. Pistepilvien tapauksessa malli voi olla esimerkiksi taso, pallo tai sylinteri, tai tarkemmin ottaen niiden parametrisoidut yhtälöt. Käytännössä algoritmi toimii seuraavasti:

1. Etsitään satunnaisesti pienin osajoukko, jolla mallin ehdot täyttyvät (RAN)
2. Parametrisoidaan malli vastaamaan näitä pisteitä
3. Etsitään koko datajoukosta pisteet, jotka ovat konsistentteja mallin kanssa, eli tyypillisesti enintään virhetoleranssin päässä mallista (nk. *consensus set*)
4. Malli voidaan uudelleenestimoida koko joukolla
5. Aikaisemmat kohdat toistetaan määritellyn iteraatiokertojen lukumäärän verran
6. Kaikista tuotetuista malleista valitaan se, joka vastaa parhaiten määritettyä hyvyysehtoa

RANSAC on suosittu menetelmä pistepilvien käsittelyyn erityisesti sen takia, että se on virhetolerantti [46]. Data voi sisältää runsaasti poikkeavia havaintoja ilman, että se vaikuttaa merkittävästi kappaleen tunnistamiseen. Tämä on hyödyllinen ominaisuus ajatellen tyypillistä reaalimaailmasta otettua pistepilveä, jossa poikkeavia tuloksia voi tulla monista lähteistä. Kameroiden epätäydellisyys ja mittausepätarkkuus aiheuttavat jo lähtökohtaisesti virhettä, mutta myös heijastavat pinnat, pienet leijuvat kappaleet ilmassa tai monimutkaiset ja hankalasta kulmasta kuvatut muodot vääristävät tuloksia.

4.5.1 Pistepilven segmentointi

RANSAC itsessään ei vielä riitä laajan pistepilviaineiston analysointiin, koska se löytää ainoastaan yhden pistejoukon, joka muodostaa yhden kappaleen.

Koko pisteavaruus koostuu kuitenkin lähtökohtaisesti useista kappaleista, jotka voivat olla erikokoisia ja erimuotoisia. Tämän vuoksi täytyy RANSAC ajaa useaan kertaan. Jokaisella suorituskerralla löydettyyn malliin kuuluvat pisteet poistetaan alkuperäisestä pistejoukosta, jotta samaa kappaletta ei löydetä uudestaan. Näin jatketaan, kunnes etsittyjä muotoja ei enää löydetä. RANSAC:a käytettäessä on huomioitava, että se antaa epäoptimaalisen lopputuloksen asetetulla todennäköisyydellä. Tämän vuoksi on löydetyt mallit syytä sovittaa uudelleen [30]. Tällöin mallia hienosäädetään sen läheisyydestä löytyvien pisteiden avulla, jolloin löydetään optimaalinen malli.

Tämän tutkimuksen puitteissa kokeiltiin sylinterimuodon etsintään myös menetelmää, jossa ajatuksena on segmentoida koko pisteavaruus alkeellisiin muotoihin. Näin saataisiin parempi kuva siitä, mitkä pisteet eivät todenäköisesti kuulu etsittyihin kappaleisiin. Tästä lähtökohdasta sovellettiin Schnabelin vuoden 2007 tutkimusta. Tutkimusryhmä jakoi testiaineistonsa menetelmällään viiteen erilaiseen muotoon: tasoihin, palloihin, sylintereihin, lieriöihin ja toruksiin käyttämällä kullekin soveltuvaa matemaattista mallia. Algoritmi perustuu ideaan, että samankokoiset muodot etsitään kokojärjestyksessä suuresta pieneen. Lopputuloksena he saivat koko testimallinsa jaettua noin 20 sekunnissa alkeellisiin muotoihin. [30]

4.6 Klusterointi

Pistepilvien käsittely on huomattavasti tehokkaampaa ja virheitä tulee vähemmän, kun käytetyt algoritmit voidaan jakaa tarpeeksi pieneen pistejoukkoon. Tiedetään, että lähekkäiset pisteet ovat todennäköisesti osa samaa kappaletta. Näiden toistaan lähellä olevien pisteiden löytäminen tapahtuu yksinkertaisimmillaan kd-puuta hyväksikäyttäen etsimällä kullekin pisteelle naapuripisteet, jotka ovat määrätyn rajaetäisyyden sisäpuolella. Jotta löydetyt klusterit eivät muodostu liian laajoiksi, rajataan yhteen joukkoon kuuluvien pisteiden määrää. Tämä algoritmi on muodollisemmin nimeltään euklidinen klusterin erottelu (Euclidean Cluster Extraction) [44]. Lopputuloksena koko kohdepilvi on rajattu likimain samankokoisiin pistejoukkoihin, joita on helppo käsitellä.

4.7 Kinect Fusion

Kinect Fusion on erityisesti Kinectille kehitetty, mutta myös muille syvyyskameroille yleistettävä algoritmi, jossa liikkuvan kameran avulla voidaan muodostaa kuva ympäristöstä. Algoritmi seuraa kameran liikettä käyttämällä

Iterative Closest Point -menetelmää (ICP) [50]. ICP:tä käytetään usein mallin transformaatiomatriisin laskemiseen silloin, kun likimain sama malli on kuvattu kahdesta eri kohdasta. Kinect Fusionissa ICP:n avulla muodostettua kuuden vapausasteen transformaatiomatriisia käytetään Kinectin sijainnin ja orientaation laskemisessa. [51]

Kinect Fusionin etuna on sen reaaliaikaisuus. Vastaavat menetelmät ennen Kinect Fusionia olivat joko liian hitaita tai sitten ne pystyivät ainoastaan seuraamaan kameran liikettä, mutta eivät rakentamaan ympäristön mallia. Toinen hyödyllinen ominaisuus on sen sietokyky liikkuville kohteille. Jonkin aikaa paikallaan pysyvät kohteet aiheuttavat kuitenkin algoritmille ongelmia, kuten aiheuttavat myös koko näkökentän peittävät kappaleet. Kolmas hyödyllinen ominaisuus on automaattinen taustan erottelu liikkuvista kohteista. [51]

Luku 5

Prototyypin toteutus

Luvussa 3 valittiin tutkittavaksi menetelmäksi letkun havaitsemisen pistepilvestä. Tässä luvussa käsitellään prototyyppitoteutusta käytännön tasolla ja kerrotaan, kuinka erilaisia pistepilvitekniikoita on hyväksikäytetty laitteiston toteuttamisessa. Käytetyt menetelmät esitellään vain pintapuolisesti: tarkemmin niistä voi lukea luvusta 4 Pistepilvien teoriaa. Lisäksi tässä luvussa pohditaan käytännön ongelmia, joita prototyyppiä toteutettaessa tuli esille ja pohditaan vaihtoehtoisia tapoja toteuttaa valittu menetelmä.

Seuraavaksi esitellään prosessi panostuksen seuraamiseksi. Ennen panostusta otetaan niin sanottu referenssikuva koko panostusalueesta. Panostuksen aikana otetaan useita pistepilvikuvia panostusprosessin etenemisestä ja varsinkin letkun liikkeestä. Kuvia otetaan mahdollisesti usealla kameralla ja monesta suunnasta. Panostuksen jälkeen referenssikuvaan yhdistetään manuaalisesti räjäytyskuvio, eli koordinaatit, joissa reikien pitäisi sijaita. Tämän jälkeen käsitellään panostuksen aikana otetut kuvat ja analysoidaan, missä kohtaa letku on kuvassa. Havaituista letkun osista etsitään referenssikuvaan vertaamalla ne, jotka ovat lähinnä seinää ja panostettavia reikiä. Niitä lähin reikä tulkitaan panostuksen kohteeksi kullakin ajanhetkellä. Tämä tieto yhdistetään panostuslaitteen aikaleimoihin ja emulsion määriin, jolloin tiedetään, paljonko räjähdysainetta menee kuhunkin reikään.

Edellä mainittu prosessi on alkuperäinen idea, jolla panostusprosessin seuraamista lähdettiin kehittämään. Käytännössä prototyyppi ei pysty toteuttamaan kaikkia prosessin vaiheita. Tausta pystytään poistamaan pistepilvestä ja sylinterit pystytään paikantamaan. Letkun muodostusheuristiikkaa ei ehditty projektin puitteissa toteuttaa. Prototyyppitoteutus tekee myöskin oletuksen, että kamera ei liiku referenssikuvan ottopaikasta. Todellisuudessa algoritmin täytyy pystyä tunnistamaan, mitä osaa referenssikuvasta tarkkaillaan. Puutteista huolimatta luvussa esitellään myös suunnitellut ominaisuudet.

5.1 3D-kamerat

Konenäkösovellusta varten tarvitaan keino tuottaa pistepilvi analysointia varten. Teoriassa on mahdollista käyttää stereokuvaa jonkinlaisen pistepilven tuottamiseksi. Stereokuvan käyttäminen on kuitenkin laskennallisesti vaativa operaatio, joka vaatii piirteiden eristämistä ja vertailua kuvien välillä [52]. Parempi tulos, joka toimii myös huonommassa valaistuksessa, saadaan aikaan jollakin kaupallisella 3D-kameralla. Tässä yhteydessä puhuttaessa 3D-kamerasta tarkoitetaan nimenomaan syvyyskameroita, eikä esimerkiksi filmiteollisuudessa käytettyjä stereokameroita, jotka simuloivat ihmisen näköä [53].

3D-kamerateknologia on edennyt paljon viime vuosina. Kuluttajille teknologian teki tutuksi ensimmäinen Kinect, joka julkaistiin marraskuussa 2010 [54]. Tämän jälkeen lisätyn todellisuuden ja 3D-tulostuksen saattelemana 3D-kamerat ovat arkipäiväistyneet, mutta käytännön sovelluksia on vähemmän. Teollisuudessa 3D-kameroita on käytetty pidempään, mutta nämä ovat tyypillisesti huomattavasti kalliimpia ja kookkaampia laitteita kuin kuluttajakäyttöön tarkoitetut laitteet [55]. Tässä projektissa kameran koolla on sen sijoituspaikka huomioiden merkitystä. Lisäksi halutaan etsiä suhteellisen edullista ratkaisua ongelmaan, joten tarkastelussa on lähinnä kuluttajahinnoissa liikkuvia 3D-kameroita. Nämä kamerat perustuvat kahteen teknologiaan: Time of Flight (TOF) ja rakenteiseen valoon (engl. *structured light*). Tässä esitelläänmolemmat teknologiat, sekä esimerkkilaitteita, jotka hyödyntävät näitä teknologioita.

5.1.1 Time of Flight

Yleisesti Time of Flight -teknologia perustuu nimensä mukaisesti lentoajan mittaamiseen lähteestä havaitsijaan. Kun käytettävän aallon tai partikkelin nopeuden tunnetaan, voidaan laskea kuinka pitkän etäisyyden se kulki matkallaan. Mittauskohteena voidaan käyttää esimerkiksi kappaletta, ääniaaltoa tai valoa. 3D-kameroissa käytetään usein infrapunavaloa etäisyyden mittaamiseen, sillä ihminen ei näe sitä, eikä se myöskään aiheuta ylimääräistä häiriötä ympäristölle. Valon nopeus tunnetaan hyvin eri väliaineissa, joten etäisyyden laskenta on suhteellisen yksinkertainen laskutoimitus tapauksessa, jossa mitataan yhden suunnatun valonsäteen kulkuaika. TOF-kameroissa yhdellä pulssilla valaistaan koko kuva, toisin kuin esimerkiksi LIDAR-skannereissa, joissa jokaisella pulssilla tuotetaan yksi kuvan piste. TOF-kameroissa etäisyyden mittaus tehdään moduloimalla

	Kinect 2.0[58]	**Structure Sensor**[59]	**Structure Core**[60]
Resoluutio	512x424	640x480	2x 1280x960 640x480
FoV	70°x60°	58°x45°	59°x45,8° 159°diagonaali
Päivitystaajuus	30Hz	30/60Hz (VGA/QVGA)	30/60Hz
Etäisyys	0,5-4,5m	0,4-3,5m	0,3-5,0m
Tarkkuus	Tuntematon, mutta parempi kuin Structure Sensor [61]	0,5mm/40cm 30 mm/3 m	0,17% /1m 0,55% /3,5 m
Mitat	24,9 cm x 6,6 cm x 6,7 cm	11,9 cm x 2,8 cm x 2,9 cm	10,1 cm x 2,0 cm x 1,41 cm
Paino	1,4 kg	95 g	39 g
Hinta	109,90€ (+42,90€ adapteri)[62]	$349 (+ $ 29 virtalähde)[63]	Ei julkaistu 11.11.2017

Taulukko 5.1: Vertailtavien kameroiden tietoja

valonlähdettä ja tarkkailemalla heijastuvaa valoa. Heijastuksessa tapahtuu vaihesiirtymä, jota vertaamalla lähdevaloon voidaan etäisyys laskea. [52]

5.1.2 Rakenteinen valo

3D-kamerat, jotka toimivat rakenteisella valolla projisoivat tunnetun kuvion skannattavan kohteen pinnalle. Kuten TOF-kameroissa, rakenteinen valo toimii yleensä ihmiselle näkymättömällä infrapuna-aallonpituudella. Pinnalle heijastettu kuvio näkyy vääristyneenä eri kuvakulmista ja erilaisilla pinnoilla, jolloin vertaamalla sitä alkuperäiseen kuvioon voidaan laskea etäisyys kohteesta. Yleinen käytetty kuvio koostuu yksinkertaisesti viivoista, mutta esimerkiksi alkuperäinen Kinect heijastaa pinnalle satunnaisen pistekuvion [56]. [57]

5.1.3 Kameran valinta

Projektissa käytettäväksi kameraksi valittiin lopulta Kinect 2.0. Kamera yhdistetään Windows-tietokoneeseen. Muutama vuosi sitten kameroita

oli markkinoilla enemmän, mutta parina edellisenä vuonna ne ovat lähes hävinneet markkinoilta. Kinect 2.0 on käytännön olosuhteisiin hieman kookas vaihtoehto, mutta toisaalta prototyyppitoteutusta tehtäessä koko on toisarvoinen kriteeri. Sen sijaan Kinect 2.0 on markkinoiden ylivoimaisesti paras tarkkuudessa hintaluokassaan. Markkinoiden toinen laite Structure Sensor on huomattavasti pienempi, mutta se ei pärjää tarkkuudessa tai näkökentässä Kinect 2.0:lle. Näistä tarkkuutta tarvitaan sylintereiden luotettavaan tunnistukseen. Laaja kuvakulma puolestaan on tärkeä, kun mietitään lopullista käyttötarkoitusta: letkun täytyy olla koko ajan kameran näkökentässä. Kinect 2.0:n kanssa työskennellessä huomionarvoista on lämpötilan vaikutus sen toimintaan. Kamera antaa muuttuvaa poikkeamaa todellisesta etäisyysarvosta, kunnes se on lämmennyt tarpeeksi. Tämä tapahtuu noin 20 minuutin kohdalla, jonka jälkeen kameran tuulettimet käynnistyvät ja vakauttavat lämpötilan. [61]

Huomionarvoista kameran valinnassa on, että kamerateknologia kehittyy jatkuvasti. Projektin alkupuolella Kinect oli hinta-laatusuhteeltaan merkittävästi parempi kuin kilpailijansa. Myöhemmin erityisesti Structure on kehittänyt teknologiaansa eteenpäin ja tarjoaa Structure Core -kameraa, joka on määrittelyidensä perusteella ylivoimainen lähes kaikilla mittapuilla verrattuna sekä Structure Sensoriin että Kinect 2.0:aan. Marraskuussa 2017 sitä ei vielä myydä verkkokaupassa, mutta kehittäjäversioita on tarjolla. Niiden hinta on kuitenkin useita tuhansia euroja, mikä tekee siitä toistaiseksi liian kalliin vaihtoehdon. [60]

5.1.4 Kameran sijoittaminen

Kuvasta 2.4 nähdään, että tunnelinperä voi olla hyvinkin ahdas. Teoriassa sivuilla on tilaa kameralle, mutta ne pitäisi pultata seinään kiinni hyvin lähelle tunnelin perää, jotta panostuskori ei peitä niitä. Tämä puolestaan vaikeuttaisi varsinaista tunnistusoperaatiota, koska sivulta kuvattuna kameran on vaikea erottaa vierekkäisiä reikiä toisistaan. Lisäksi sivulta kuvattuna mahdolliset tunnelinperän kielekkeet voivat tulla eteen, kuten nähdään kuvasta 3.2. Toinen vaihtoehto on pistää kamera suoraan panostajaan kiinni, mutta panostajan liike ja vaadittavat liitännät tekevät tästä hyvin epäkäytännöllisen ratkaisun. Käytännössä ainoa paikka johon kameran voi sijoittaa siten että se näkee panostettavan alueen ilman esteitä on panostuskori itsessään. Jotta kuvakulma on paras mahdollinen, tulee kamera sijoittaa panostuskorin ylälaitaan hieman alaviistoon. Tällöin panostaja ei tule liiaksi eteen, ja panostettava reikä on hyvin nähtävissä. Korin alalaitaan sijoitettaessa panostaja saattaa helposti tulla kameran eteen, sillä pohjareikien panostus tehdään usein ilman panostuskoria.

Kuva 5.1: Leveä panostuskori. Huomionarvoista on, että panostuskorin katto ei ulotu etureunaan asti.

Kaiken kattavaa sijoituspaikkaa kameralle on vaikea määrittää, sillä panostuskoreja on erilaisia. Kaksi esimerkkiä on nähtävissä kuvasta 5.1 ja 5.2. Huomataan, että molemmissa panostuskoreissa ei edes ole etureunaan ulottuvaa kattoa, jonka etureunaan kameran voisi sijoittaa. Kameran sijoittamisessa panostuskoriin on myös yleisiä ongelmia. Kameran ollessa lähellä seinää ei sen ole mahdollista nähdä koko kuvattavaa aluetta. Riittävän suuri kuvausalue on välttämätön, sillä menetelmämme perustuu referenssikuvaan vertaamiseen. Liian pienellä kuvausalueella ei alueen tunnistaminen ole mahdollista. Tätä ongelmaa voidaan yrittää kiertää useammalla eri suuntiin kohdistetulla kameralla, joiden yhteiskuva tuottaa todellisen sijainnin.

5.2 Letkuntunnistusalgoritmi

Putken tunnistus on tärkein tekninen ongelma, jota tässä diplomityössä ratkaistaan. Tätä varten kehitettiin algoritmi, joka referenssipilveä ja työn aikaista pilveä verraten osaa päätellä, missä letku kullakin ajanhetkellä sijaitsee. Yksinkertaistaen algoritmi koostuu seuraavista vaiheista:

1. Referenssikuvan ottaminen
2. Työnaikaisen kuvan eroavaisuuksien löytäminen referenssikuvasta
3. Eroavaisuuksien klusterointi

Kuva 5.2: Pieni panostuskori.

4. Sylinterimuotojen segmentointi klustereista
5. Letkumuodon päätteleminen sylinterimuodoista

Tässä osiossa esitellään jokainen vaihe tarkemmin ja perustelut juuri näiden menetelmien käyttämiselle. Kehityksessä on käytetty apuna Point Cloud Library -C++ kirjastoa, joka sisältää kattavasti valmiita algoritmeja pistepilvien käsittelyyn. Kaikkea ei siis ole tarvinnut ohjelmoida alusta lähtien itse.

5.2.1 Referenssikuva

Kehitetty algoritmi olettaa, että ennen panostusta on otettu referenssikuva koko panostusalueesta. Referenssikuva tarvitaan, jotta on mahdollista löytää pistepilvistä ne kohdat, joissa letku voi mahdollisesti sijaita. Jos jokin ympäristön piirre sijaitsee sekä referenssikuvassa että panostuskuvassa, voidaan se eliminoida potentiaalisista letkupisteistä. Referenssikuvaa voidaan potentiaalisesti käyttää myös kameran paikantamiseen.

Toinen vaihtoehto referenssikuvan lisäksi olisi muodostaa kuva panostusalueesta sitä mukaa kuin työtä tehdään. Tämä on teoreettisesti mahdollista esimerkiksi Kinect Fusion -menetelmällä. [51]. Algoritmilla on kuitenkin rajoitteensa, minkä takia se päätettiin jättää hyödyntämättä tässä projektissa. Merkittävin ongelma on panostajan liikkuminen kuvausalueella. Algoritmi kykenee erottamaan liikkuvia kohteita, mutta panostaja ei välttämättä liiku tarpeeksi tätä varten. Tämä aiheuttaa panostajan hahmon sekä letkun ilmestymisen referenssikuvaan. Algoritmia täytyisi siis jatkokehittää. Kehitys tuskin olisi mahdotonta, sillä panostusprosessin etenemistä ei tarvitse seurata reaaliajassa. Sen sijaan referenssikuva voidaan esiprosessoida kaikesta syntyneestä datasta jälkikäteen, jolloin liikkuvat osat voidaan suodattaa pois. Etuna yhden referenssikuvan käyttämiselle olisi toisaalta huomattavasti tiheämpi pistepilvi, mikä on todennäköisesti välttämätön tarvittavan tunnistustarkkuuden saavuttamiseksi. Tälle projektille varatuilla resursseilla ainoaksi vaihtoehdoksi jää kuitenkin vain yhden pysyvän referenssikuvan käyttäminen.

5.2.2 Pistepilvien erottelu

Ei ole tarkoituksenmukaista analysoida koko pistepilveä sylinterimuotojen havaitsemiseksi. Tätäkin vaihtoehtoa tutkittiin, mutta kävi ilmi, että käytetyn 3D-kameran tarkkuus tai pistetiheys eivät riittäneet muotojen tarkkaan tunnistamiseen. Schnabelin ryhmän tutkimuksessa, jossa näin oli tehty onnistuneesti, annettiin esimerkkinä "pieni"sylinteri, joka piti sisällään 1366 pistettä [30]. Dilplomityön tapauksessa pienimmät muodot - lyhyet sylinterinpätkät - voivat olla vain joitain kymmeniä pisteitä. Ongelma havainnollistui, kun tasoja yritettiin suodattaa RANSAC:lla pois pistepilvestä: tasojen paksuuden toleranssin täytyi olla suhteellisen suuri, jotta kaikki tason pisteet saatiin mukaan. Tästä seurasi, että satunnaisesti myös letku havaittiin tasona, sillä testeissä käytetyn letkun halkaisija oli alle 2cm. Tämän vuoksi päädyttiin käyttämään referenssikuvaa ja prosessoimaan sen avulla eroavaisuudet kahdesta pistepilvestä.

Edellä havaittiin, että kaikkien muotojen havaitseminen koko pistepilvestä on liian epävarma prosessi. Sen lisäksi se on myös hyvin hidas. Schnabelin tutkimuksessa kului kymmeniä sekunteja, kun mallista haettiin kaikki muodot. Tämän projektin vähemmän optimoidulla segmentointialgoritmilla kului yhden pistepilven analysointiin jopa minuutti. Yhden panostuskerran aikana pistepilvidataa syntyy useiden satojen kuvien verran pienelläkin näytteenottotaajuudella. Koko analyysiin voisi siis mennä kokonainen päivä. Pysyvän ympäristön suodattaminen pistepilvestä nopeuttaa segmentointivaihetta pienentämällä analysoitavan datan määrää, mutta erottelualgoritmi

voi itsessään olla hidas. Joskus se voi olla jopa hitaampi kuin koko pistepilven segmentointi. Havaittiin, että panostuskuvan ja referenssikuvan erot löytävän algoritmin kesto on vahvasti riippuvainen pisteiden määrästä. Kun pistepilvissä oli kaikki pisteet, eli noin 200 000, kesti algoritmin suorittaminen useita minuutteja. Kun pistepilvet harvennettiin alle 100 000 pisteeseen kesti erottelu enää sekunteja. Harventamisen jälkeen pistepilvien erottelu suoritettiin 1cm lehtikoolla, joka oli sama kuin harventamisessa käytetty. Tämä tuotti suhteellisen luotettavasti pistepilvien erotuksen. Satunnaisesti referenssipilvessä oli häiriöpisteitä juuri sillä alueella, missä putki liikkui. Tällöin jokin tärkeä piirre varsinaisesta työkuvasta katosi, koska algoritmi etsi kaikki läheiset pisteet ja häiriöpisteiden tapauksessa löysi niiden läheltä myös letkun muodostavia pisteitä. Tämä tapahtui kuitenkin niin harvoin ja yleensä vain ennen kameran lämpenemistä, että ei katsottu tarpeelliseksi käsitellä referenssikuvaa häiriöiden poistamiseksi.

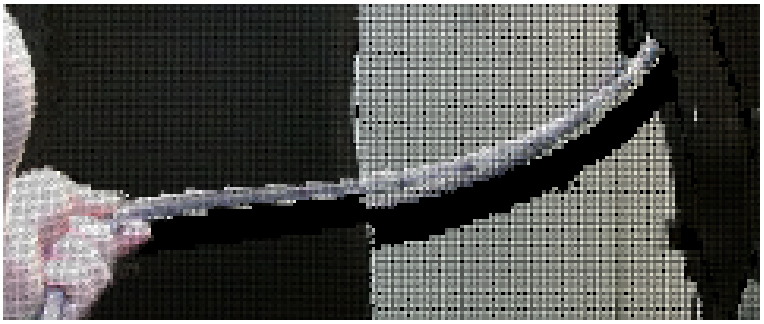
Erotuksen jälkeen pistepilven tiheys päätettiin palauttaa alkuperäiseksi kaiken saatavilla olevan datan hyödyntämiseksi sylintereiden etsimisessä. Tämä onnistuu muodostamalla kd-puu alkuperäisestä pistejoukosta ja suorittamalla jokaiselle sen pisteelle sädehaku kd-puun avulla. Etsintäsäteenä käytettiin arvoa 2cm, joka riittää varmasti kaikkien harvennetun pistepilven pisteiden havaitsemiseen. Koska pistepilven harventaminen tapahtuu vokseliverkon avulla, tiedetään että maksimietäisyys harvennetun pisteen ja jonkin vokselin sisällä olevan alkuperäisen pisteen välillä on $\sqrt{3}$ cm $\approx$ 1,73 cm. Tämä on siinä tilanteessa, jossa painopiste on lähellä yhtä vokselin kulmaa ja jokin alkuperäisen pilven piste on lähellä täysin vastakkaista kulmaa. Pelkkä sädehaku tuottaa luonnollisesti myös duplikaattipisteitä, joten lopuksi pistepilvestä poistetaan ylimääräiset pisteet.

5.2.3 Sylinterien tunnistaminen

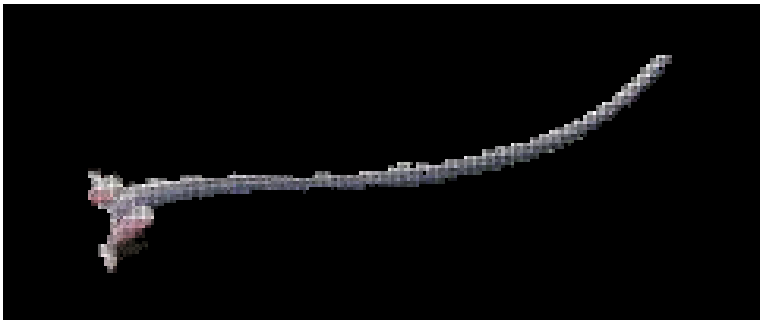
Seuraava vaihe letkuntunnistuksessa on pistepilven segmentointi sylintereihin. Tämä tapahtuu RANSAC-algoritmia hyväksi käyttäen. Sitä varten tarvitsee vielä käsitellä erotuksesta jäljelle jääneitä pisteitä. Ensimmäinen operaatio pistepilvelle on klusterointi käyttäen euklidista klusterinhakua. Klusterointi tehdään, jotta RANSAC ei turhaan löytäisi sylinteriin pisteitä, jotka eivät muodosta yhtenäistä kappaletta. PCL:n sylinterimalli ei sisällä päätepisteitä, vaan ainoastaan akselin ja säteen. Sylinteri on siis näennäisesti äärettömän pitkä ja kaikki pisteet, jotka täyttävät akseli- ja säde-ehdon katsotaan kuuluvaksi sylinteriin. Klusterointialgoritmi määritetään niin, että kaikki klusterin pisteet ovat enintään 2 cm päässä lähimmästä naapuristaan. Lisäksi klusterin täytyy koostua 20 - 25 000 pisteestä. Jokainen klusteri

tallennetaan omaan pilveensä, joista jokaiselle suoritetaan sylinterihaku erikseen.

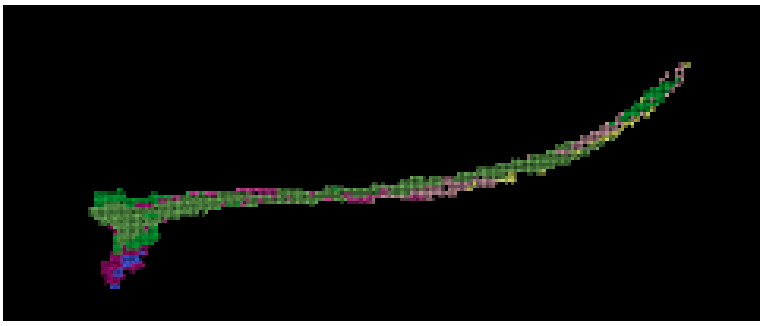
Sylinterihaussa klusterille estimoidaan normaalit RANSAC:a varten. Niitä käytetään sylinterimallin yhteydessä määrittämään sylinterin ulkopintaa. Estimointialgoritmi käyttää hyväkseen pistettä ja sen viittä lähintä naapuripistettä arvioinnissaan. Normaalien estimoinnin jälkeen etsitään kaikki sylinterit, jotka täyttävät valitut kriteerit. Tätä varten muodostetaan sylinterimalli, johon RANSAC vertaa pisteitä. Testeissä käytetty puutarhaletku on säteeltään 6,5 mm. Etsityn sylinterin keskimääräiseksi säteeksi asetetaan kuitenkin mittausepävarmuudesta johtuen 10 mm. Näin varmistetaan, että kaikki sylinterin pisteet tulevat mukaan sylinteriin. Tämä täytyy tehdä, koska käytetty kameran mittausepätarkkuus on huomattava verrattuna putken säteeseen. Sylinterimalli täydennetään vielä niin, että sylinteriin kuuluvat kaikki pisteet, joiden etäisyys sylinterin keskipisteestä on minimissään 6 mm ja maksimissaan 14 mm. Näihin lukuihin lasketaan vielä lisäksi RANSAC:lle annettu toleranssi 5 mm, jolloin kokonaisväliksi tulee 1 mm - 19 mm. Mainittakoon, että todellisuudessa panostusletku on hieman puutarhaletkua paksumpi, noin 30 - 50 mm [64]. Kaksi muuta parametria RANSAC:lle ovat iteraatiokierrosten lukumäärä 2500 sekä todennäköisyys, että löydetty sylinteri on paras mahdollinen 0.95. Tämä algoritmi suoritetaan niin monta kertaa, että se löytää kaikki sylinterit, joissa on vähintään 15 pistettä. Tämä lukumäärä on kokeilulla löydetty tulos, joka riippuu täysin käytetyn kameran tarkkuudesta ja pistetiheydestä. On huomioitava, että kameran arvioitu tyypillinen etäisyys panostuksesta on vain noin metri. Näin lähellä pistetiheys on vielä suhteellisen suuri ja minimipistemäärää voidaan nostaa suuremmaksikin kuin tässä käytetty 15 pistettä. Aina kun algoritmi löytää sylinterin, poistetaan sen sisältämät pisteet klusterin pistejoukosta. Tällä tavalla pisteet eivät päädy useampaan kuin yhteen sylinteriin ja kokonaispistemäärä pienenee jatkuvasti taaten ohjelman päättymisen. Kuvassa 5.3 näytetään yksi esimerkki testipistepilvestä, sekä algoritmin tuottama näkemys siitä, millaisista sylintereistä pistepilvessä esiintyvä letku muodostuu.



(a) Alkuperäinen pistepilvi



(b) Pistepilvi, josta on suodatettu tausta



(c) Letku sylinterintunnistuksen jälkeen. Sisäkkäiset sylinterit on esitetty eri väreillä.

Kuva 5.3: Letkuntunnistuksen kolme vaihetta

Edellä kuvattiin lopullinen toteutus, joka toimi testeissä parhaiten. Projektin aikana kokeiltiin myös tuloksettomasti kahta hieman monimutkaisempaa tapaa, joita on onnistuneesti käytetty paremmissa olosuhteissa. Ensimmäisessä tavassa pyrittiin koko pistepilvi segmentoida erikokoisiksi kappaleiksi: tasoiksi, palloiksi ja sylintereiksi. Tämä ei toiminut johtuen kahdesta syystä. Ensinnäkin, alkuperäisessä Schnabelin tutkimuksessa oli käytetty huomattavasti tiheämpiä malleja kuin mitä Kinect pystyy tuottamaan [30]. Kinectin tuottaman mallin sylinterit saattavat olla vain

muutamia pisteitä, kun taas tutkimuksessa pieni sylinteri oli yli tuhat pistettä. Toisekseen kameran tarkkuus heikkenee, mitä kauemmaksi kamerasta mennään. Esimerkiksi tason toleranssiksi täytyi asettaa useita senttimetrejä, ennen kuin kaukainen seinä saatiin poistettua yhdellä haulla. Pienemmällä toleranssilla seinä löytyi usein, mutta siitä poistui vain pieni kaistale jokaisella iteraatiolla. Lisäksi useiden iteraatioidenkin jälkeen pistepilveen jäi vielä paljon yksittäisiä pisteitä seinän läheisyyteen.

Toinen kokeiltu tapa liittyi myöskin Schnabelin tutkimuksen pääasialliseen tulokseen. Ideana siinä on, että samanmuotoisia kappaleita etsitään isosta kappaleesta pieneen siten, että esimerkiksi ensiksi pistepilvestä poistetaan isot sylinterit, sitten keskikokoiset ja sitten pienet. Tämä tapa osoittautui hankalaksi toteuttaa käytännössä tuntematta tarkkoja parametreja, joita tutkimuksessa käytettiin. Lisäksi mallin epätarkkuudesta johtuen, pienikin sylinteri rekisteröityi helposti vähän suuremmaksi ja poistui näin pistejoukosta ennen aikojaan. Vastaavia tekniikoita tuskin kannattaa lähteä toteuttamaan ilman merkittävästi parempaa 3D-kameraa.

Varsinainen projekti loppui, kun prototyyppi saatiin tähän vaiheeseen. Tulevissa osioissa kuvataan kuitenkin valmiita ideoita siitä, miten löydettyjä sylintereitä voisi käyttää hyväksi letkun tunnistamisessa ja miten tuotettu data suhtautuu muuhun valmiiseen dataan.

5.2.4 Letkuntunnistus

Sylintereiden havaitseminen on ainoastaan ensimmäinen vaihe koko letkuntunnistusoperaatiossa. Koska muutospistepilvestä ei ole eroteltu muita muotoja kuin sylintereitä, eikä sylintereiden kokoa ole rajoitettu, löytyy sylintereitä väistämättä myös panostajasta sekä muusta liikkuvasta ympäristöstä. Näin ollen sylintereistä pitää päätellä, mitkä niistä ovat osa letkua ja mitkä ovat muita kappaleita. Periaatteellisella tasolla voisi olla mahdollista löytää sylintereitä niin että seuraava sylinteri alkaa siitä, mihin edellinen sylinteri loppuu. Käytännössä operaatio ei ole näin yksinkertainen, sillä sylinterit ovat osittain sisäkkäin, kuten on nähtävissä kuvasta 5.3c. Kuvasta voidaan huomata myös, että liian yksinkertaisella letkukriteerillä esimerkiksi käsivarsi voitaan tulkita kasaksi letkuja. Tämän välttämiseksi ehdotetaan seuraavia kriteerejä, jotka määrittävät letkun sylintereistä.

1. Käytetään ainoastaan sylintereiden alku- ja päätepisteitä
2. Verrataan jokaista sylinteriä muihin sylintereihin ja valitaan letkukandidaateiksi ne, jotka ovat tarpeeksi lähellä toisiaan
3. Sovitetaan letkukandidaattien alku- ja loppupisteet suoralle. Ne jotka eivät osu tarpeeksi hyvin suoralle poistetaan kandidaattilistalta

4. Kandidaattipareista muodostetaan pidempiä kokonaisuuksia
5. Hylätään kandidaatit, jotka ovat vierekkäisiä yhdensuuntaisia letkuja

PCL:n sylinterimalli koostuu ainoastaan suoran yhtälöstä sekä sylinterin säteestä. Siltä puuttuu siis varsinainen alku- ja päätepiste. Alku- ja päätepisteet ovat kuitenkin välttämättömiä suunnitellulle algoritmille, joten ne täytyy laskea erikseen. On tiedossa, mitkä pisteet kuuluvat mallin sisään, joten sylinterin päätepisteet voidaan laskea projisoimalla kaikki sylinteriin kuuluvat pisteet sylinterin akselille. Lopuksi parametrisoimalla akselille projisoidut pisteet yhdestä muuttujasta riippuviksi, voidaan päätepisteet löytää etsimällä suurin ja pienin parametrin arvo.

5.3 Kameran paikannus

Letkun sijainnin selvittäminen kameran suhteen on tietenkin hyödytöntä, ellei ole tiedossa, missä kamera sijaitsee referenssikuvioon nähden. Pelkkä kameran sijainnin määrittäminen alussa ei riitä, sillä ainoa paikka, josta kamera näkee tarpeeksi - panostuskori - liikkuu jatkuvasti panostusprosessin aikana. Panostuskorin sijaintia ympäristöön nähden ei tunneta, kuten ei myöskään tunneta sen sijaintia panostusajoneuvoon nähden. Täytyy siis löytää tapa paikantaa kamera ympäristön suhteen, eli matemaattisesti sanottuna muuttaa kameran tuottama pistepilvi kamerakoordinaatistosta referenssikuvan koordinaatistoon.

Arvioitiin, että jokin Iterative Closest Point (ICP) -menetelmä voisi olla mahdollinen ratkaisu ongelmaan. Tätä ei kuitenkaan ehditty tutkia tarkemmin tai toteuttaa sitä käytännössä. Ongelmaksi ICP:n kanssa voi muodostua se, että kameran näkemä alue on hyvin pieni verrattuna koko panostuskuvion alueeseen. Kaukaa otettu pistepilvi on myös epätarkempi ja siinä on pienempi pistetiheys, kuin lähellä otetussa pistepilvessä. Tämä tarkoittaa, että millä tahansa vertailualgoritmilla pilviä toisiinsa verrattaisiin, täytyy vertailutoleranssien olla tarpeeksi suuria. Se taas voi vaikeuttaa pienempien piirteiden hyväksikäyttöä tunnistamisessa.

Tämän ongelman ratkaisu saattaisi helpottua huomattavasti, jos käytössä olisi edellä mainittu Kinect Fusion -menetelmä. Sen avulla olisi aina tiedossa kahden peräkkäisen pistepilven välinen muutos, jolloin kameran suhteellinen sijainti edellisiin pisteisiin tunnettaisiin. Lisäksi koska referenssikuva muodostetaan pala kerrallaan, tunnetaan myös kameran kulloinenkin sijainti referenssikuvioon aina. Ainoaksi jäljelle jääväksi tehtäväksi jää porausräjäytyskaavion manuaalinen yhdistäminen panostusdataan.

5.4 Tietojen yhdistäminen

Kun panostus on tehty ja letkun päät on tunnistettu, täytyy kaikki tuotettu tieto vielä yhdistää panostuslaitteistosta saatavan datan kanssa. Tätä varten täytyy ensiksi manuaalisesti määrittää, mitä katkoa oltiin panostamassa. Lisäksi poraus-räjäytyskuvio täytyy yhdistää tuotettuun referenssikuvaan. Yhdistämistä varten täytyy toteuttaa käyttöliittymä, jossa referenssikuva ja räjäytyskuvio näkyvät päällekkäin. Sen avulla kuviota voi kääntää, siirtää ja skaalata, kunnes sekä kuvion että referenssin reikien sijainnit vastaavat toisiaan.

Seuraavaksi tutkitaan, mikä reikä on lähimpänä letkun päätä kullakin ajanhetkellä. Tämä onnistuu vertaamalla letkun (tai letkun osien) päiden sijainteja reikien koordinaatteihin. Rei'istä valitaan se, josta on lyhin välimatka letkuun. Viimeisenä haetaan panostuslaitteelta tiedot emulsionkulutuksesta. Kulutuksen aikaleimat yhdistetään sillä hetkellä letkua lähimpään reikään, jolloin saadaan tieto emulsion määrästä kutakin reikää kohden.

5.5 Tiedonhallinta

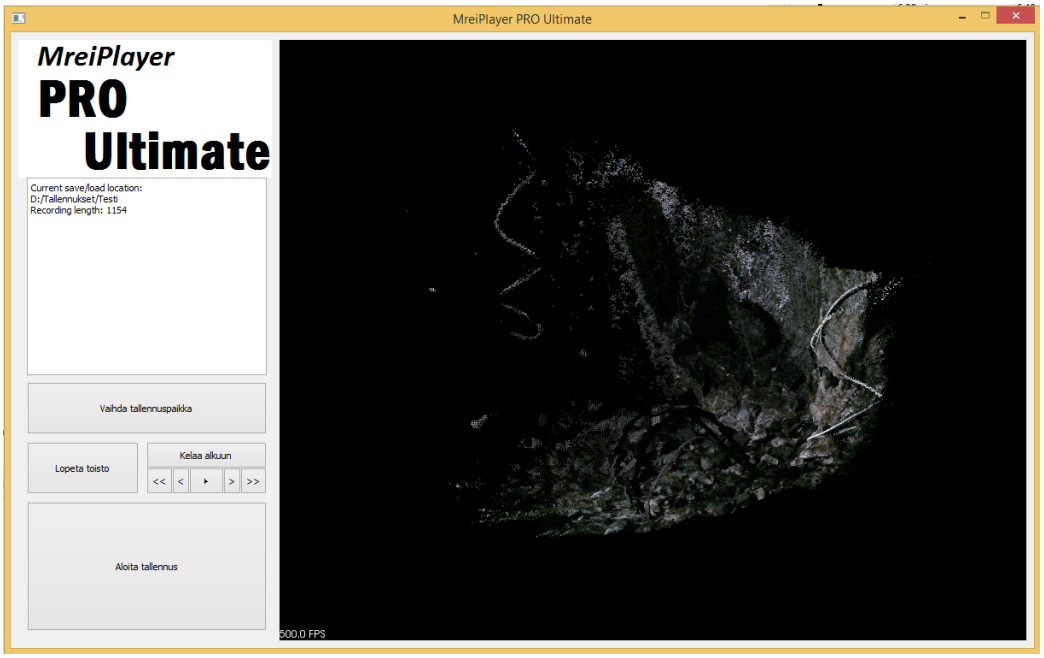
Pistepilvimenetelmille on ominaista, että ne tuottavat paljon raakadataa ja laskeminen on työlästä. Datan käsittelylle on kaksi vaihtoehtoa: laskenta voidaan tehdä joko panostusajoneuvoon asennetulla tietokoneella tai raakadata voidaan ensiksi siirtää toiseen ympäristöön. Optimitilanteessa panostusajoneuvoon ei tarvitse asentaa tehokasta laskentakonetta, vaan koneen täytyy ainoastaan olla tarpeeksi tehokas kuvaamaan prosessi ja tallentamaan tiedot. Toinen erillistä laskentaa puoltava syy on datan käsittelyyn tarvittava näyttö, jota tarvitaan räjäytyskuvion asemoimiseen. Tunnelissa se olisi jatkuvassa likaantumisvaarassa.

Vaihtoehdoiksi jää tiedon reaaliaikainen siirtäminen verkkoyhteyden avulla tai panostuksen jälkeen kovalevyn tai muistitikun toimittaminen laskentakoneen käyttöön. Nykyään tiedonsiirto alkaa olla mahdollista myös rakenteilla olevissa tunneleissa, mutta yhteydet ovat vielä kovin hitaita [65]. Hitaalla yhteydellä ei ole järkevää siirtää näin suurta määrää dataa. Toisaalta siirreltävä tiedonsiirtoväline ei ole käytännöllinen ratkaisu. Panostuslaite ei välttämättä poistu kovin usein tunnelista, joten tiedonsiirtoa ei voida hoitaa ainoastaan silloin, kun hyvät verkkoyhteydet ovat saatavilla. Tämän vuoksi jonkinlainen esiprosessointi lienee tarpeellista.

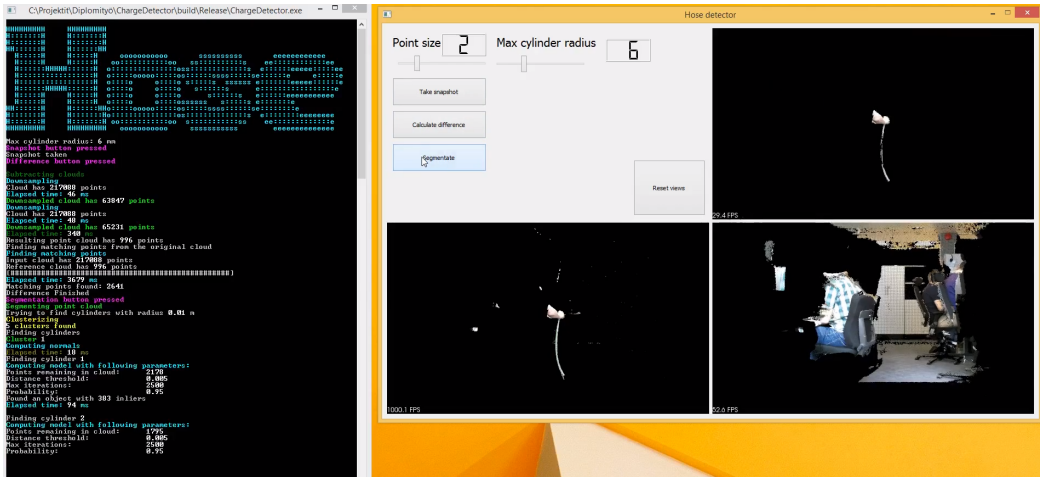
5.6 Virheidenkäsittely

Panostuksessa olosuhteet tuskin ovat täydelliset ja on mahdollista, että ei ole aina tiedossa, missä kohtaa panostusletku on siinä kohtaa, kun panostuslaite pumppaa emulsiota reikään. Tämän vuoksi on tärkeää, että prosessi selviää myös ongelmatilanteista tarpeeksi varmasti.

Ensimmäinen huomioitava asia on mahdollinen näkyvyyden puuttuminen. Täydellistä peittoa on vaikea estää, ellei panostaja erityisesti kiinnitä siihen huomiota. Kuitenkin jos vain muutaman panostuksen aikana letku ei ole ollut näkyvissä, on mahdollista tehdä arvioita panostettavasta reiästä muun datan perusteella. Voidaan esimerkiksi tehdä oletus, että panostaja on panostanut lähekkäin sijaitsevia reikiä peräkkäin. Jos kahden reiän välissä on panostamaton reikä ja näiden kahden panostuksen välissä panostettiin jokin reikä, voidaan olla hyvin varmoja että juuri panostamaton reikä jäi havaitsematta. Edellytyksenä tälle on kuitenkin, että panostamattomat reiät eivät ole liian lähellä toisiaan. Muuten on mahdotonta arvioida, missä järjestyksessä ne panostettiin. Yksittäisten panostamattomien reikien kohdalla virheet voidaan arvioida yksi yhteen.



Kuva 5.4: MreiPlayer, ohjelma pistepilvivideon tallentamiseen ja toistamiseen



Kuva 5.5: Prototyyppiohjelmisto panostuksen seuraamiseen

5.7 Ohjelmat

Tätä diplomityötä varten toteutettiin kaksi ohjelmaa. Ensimmäinen on varsinainen panostusta seuraava ohjelma ja toinen tallentaa ja toistaa pistepilvivideota. Molemmat on ohjelmoitu C++:lla käyttäen PCL:ää. Käyttöliittymäkirjastona toimii Qt. Esimerkkikuvat ohjelmista ja niiden käyttöliittymistä on nähtävissä kuvissa 5.4 ja 5.5.

Panostusta seuraava ohjelmisto toimii siten, että ensimmäisenä otetaan referenssikuva painamalla “Take snapshot” -painiketta. Seuraavana otetaan toinen kuva letkusta ja panostajasta painamalla “Calculate difference”. Tällöin ohjelma aloittaa erotuspilven laskemisen pistepilven ja juuri otetun kuvan välillä. Viimeinen painike, “Segmentate”, suorittaa sylintereiden tunnistamisen ensiksi klusteroimalla ja sen jälkeen RANSAC:n avulla. “Reset views” -painike palauttaa kaikki kolme pistepilvinäyttöä alkuorientaatioonsa.

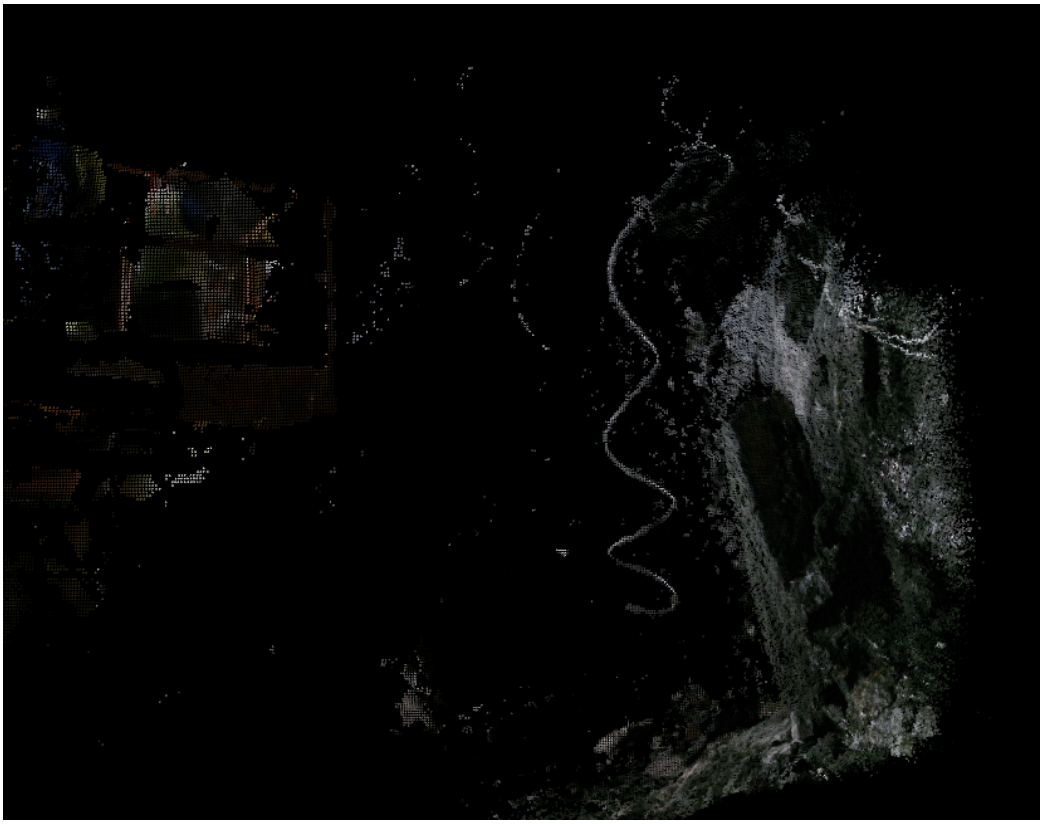
Toinen ohjelmisto MreiPlayer on pistepilvivideotallennin ja -toistin. Se tehtiin testidatan keräämiseksi tunneliympäristöstä. Tavoitteena oli selvittää, miten hyvin kamera soveltuu tunneliympäristöön sekä mahdollisesti kerätä raakaa panostusdataa oikeasta ympäristöstä. Oikean datan kerääminen ei kuitenkaan onnistunut sellaisesta kuvakulmasta, josta letkun olisi voinut tunnistaa. Ohjelma toimii tallentamalla pistepilviä yksi kerrallaan niitä sen kummemmin pakkaamatta. Kuvia otetaan kaksi kuvaa sekunnissa. Tuotettua videota voi toistaa joko reaaliaikaisesti tai käydä kuvia yksi kerrallaan läpi. Kuvat tallennetaan yhteen kansioon numeroituna ja tallennuspaikkaa voi vaihtaa.

Luku 6

Tulokset

Tässä osiossa analysoidaan työn tuloksia ja lopputuloksen soveltuvuutta alkuperäiseen tavoitteeseen. Tuloksia arvioidaan kolmelta kannalta. Ensimmäisenä tarkastellaan kaivosympäristön vaikutusta tuloksiin. Toisena arvioidaan varsinaista prototyyppiä, sen toteuttamista ja työn aikana havaittuja ongelmia. Viimeisenä esitetään arvio laitteiston nykytilasta ja jatkokehitysmahdollisuuksista sekä suositus tulevasta. Kaikki tulokset esitetään prototyypin lähtökohdasta, eli asioita tarkastellaan kamerasovelluksen kannalta.

Kuva 6.1: Töölön parkkihallin panostusta oikealta sivustalta. Letku näkyy heikosti tästä kuvakulmasta.



Kuva 6.2: Pistepilvikuva Töölön parkkihallin oikeasta sivustasta. Ainoastaan lähimmät pisteet ovat selkeästi havaittavissa. Vasemmalla on nähtävissä panostuskori, jonka kyydissä panostaja on. Letkun näköinen muoto etualalla on sytytyslanka.

6.1 Tunneliympäristö aiheuttaa haasteita

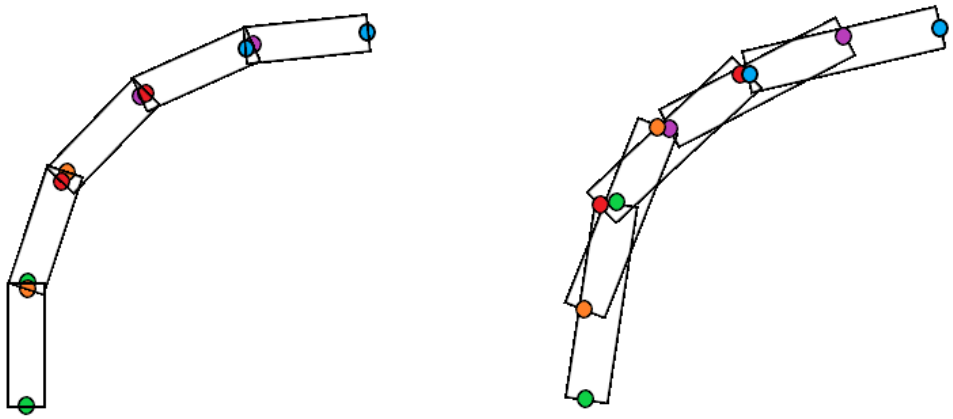
Osiossa 2.3 pohdittiin, millaisia yleisiä haasteita tunneliympäristö asettaa mahdollisille havaintomenetelmille. Kuten etukäteen arvioitiin, valaistus ei aiheuttanut merkittäviä haasteita toimintaan, vaan ympäristön valaistus riitti tavalliseenkin kuvaamiseen hyvin. Teoriassakaan valaistuksella ei pitäisi olla väliä, sillä käytetty Kinect 2.0 kamera käyttää infrapunavaloa pistepilven muodostamiseen [66]. Myöskään ympäristön lämpötilalla ei havaittu olevan merkittäviä vaikutuksia tuloksiin laitteiston lämmettyä. Suurimmat ongelmat tulivat ennakoidusti ympäristön fyysisistä rajoitteista. Kameralle ei löytynyt yhtäkään kunnollista paikkaa, josta se näkisi koko ympäristön. Kuvassa 6.1 on nähtävissä esimerkki siitä, millaisen kuvan kamera näkisi, jos se sijoitettaisiin panostusalueen sivuun. Putki on näkyvissä vain vähän

koko kuvassa ja olennainen osa jää täysin panostajan taakse. Kuvasta 6.2 puolestaan näemme, miten kamera näkee vastaavan tilanteen. Kameran etäisyysrajoitteet tulevat vastaan, ja panostuskorin sivun jälkeen pisteitä näkyy hyvin vähän. Ainostaan lähellä olevat pisteet ovat tarpeeksi tarkkoja, jotta niistä voisi tehdä minkäänlaisia johtopäätöksiä letkun sijainnista. Lisäksi huomataan, että nallien johtimet näkyvät pistepilvessä yllättävän voimakkaasti. On mahdollista, että algoritmi ei osaa tunnistaa letkua varsinkaan tilanteessa, jossa on useita johtimia sekaisin.

Toinen tunneliympäristön merkittävä ongelma on likaisuus. Töölön testaus osoittautui hyvin vaikeaksi toteuttaa käytännössä kokonaisella prototyyppilaitteistolla, johon kuuluvat tietokone, kamera ja panostusajoneuvosta lähtevä virtajohto. Testatessa täytyi jatkuvasti varoa, että käytetty kamera ja tietokone eivät likaantuisi. Panostusprosessissa ympäröivä maa oli hyvin märkää ja siellä täällä oli pieniä määriä panostusainetta, joten laitteistoa ei voinut laskea mihinkään. Kamera oli siis jatkuvasti testaajan kannossa, millä saattoi kameran heilumisesta johtuen olla jonkin verran vaikutusta kuvaustarkkuuteen. Lähellä olevat pisteet olivat kuitenkin suhteellisen tarkkoja, joten varmoja johtopäätöksiä ei tästä voi vetää.

Likaisuus vaikuttaa luonnollisesti kaikkien tunneliympäristöön sijoittuvien kamerasovellusten toimintaan. Pelkkä pöly haittaa kameran tuottaman kuvan laatua ja mikään pölyä merkittävämpi tahra voi estää näkyvyyden täysin. Tässä sovelluksessa likaisuus on erityisen suuri ongelma, sillä pienikin tahra aiheuttaa merkittävää häiriötä tuotettuun pistepilveen. Jos häiriötä syntyy liikaa, voi tunnistaminen olla mahdotonta. Panostusprosessin havaittiin olevan tunneliympäristöön nähdenkin likainen operaatio, joten kameran tai kameroiden ylläpito voi muodostua ylitsepääsemättömäksi haasteeksi. Vähintäänkin se aiheuttaa säännöllisesti ylimääräistä puhdistustyötä, mikä juurikin yritetään välttää.

Näkyvyyshaitat ja likaisuus huomioiden esitetään ainoaksi mahdolliseksi paikaksi millekään kamerasovellukselle panostuskoria. Optimaalinen paikka olisi panostuskorin katoksessa siten, että panostusta seurataan kutakuinkin yläviistosta. Kaikissa panostuslaitteissa ei kuitenkaan ole tällaista katosta, mihin kameran voisi kiinnittää. Lisäksi nykykameroista voi olla vaikea löytää sellaista, jota saisi turvallisesti kiinnitettyä panostuskoriin. Kameroita tarvitaan joka tapauksessa useita, sillä 3D-kameroiden kuvakulma ei nykyisillä tekniikoilla riitä koko työalueen näkemiseen korin työskentelyetäisyydeltä. Lisäksi samaa kohtaa olisi hyvä pystyä kuvaamaan kahdesta suunnasta, koska panostaja saattaa helposti mennä kameran ja reiän väliin. Yksi vaihtoehto tällaisissa tilanteissa, jossa näkyvyys on rajoitettu, on antaa panostalle siirtymiskehotus äänimerkin avulla.



(a) Alkuperäinen arvaus miten algoritmi löytää sylinterit letkusta. Sylinterit ovat peräkkäin niin, että niiden päätepisteet kohtaavat.

(b) Todellisuudessa algoritmi löytää useita sisäkkäisiä sylintereitä

Kuva 6.3: Vaihtoehtoisia sylinteriasemointeja letkussa

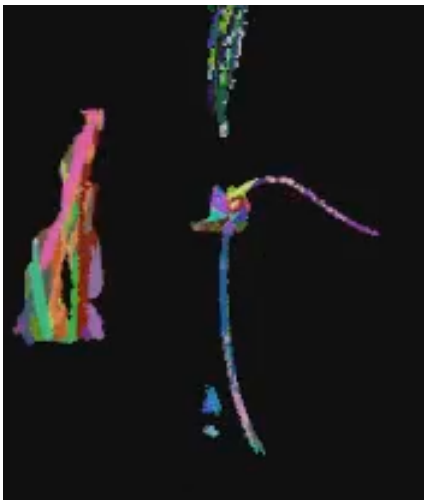
6.2 Havainnot prototyypistä

Prototyyppiä valmistettaessa kävi ilmi, kuinka hankalaa jo pelkkä letkun osien tunnistaminen on. Alkuperäinen ajatus letkun tunnistamisesta sen osioimisella sylintereiksi eli projektin loppuun asti. Huomattiin, että RANSAC toimii huomattavasti tehokkaammin, kun pistepilvi on valmiiksi jaettu tarpeeksi pieniin osiin. Tässä projektissa siihen käytettiin kahta vaihetta. Ensimmäisessä vaiheessa etsittiin erotus pistepilven ja referenssipilven väliltä. Tämä osoittautui hyvin hitaaksi operaatioksi, joka kesti jopa kymmeniä sekunteja riippuen pistejoukon koosta. Monisäikeistämällä prosessi saatiin suoritusaika pudotettua keskimäärin neljään sekuntiin. Projektin loppupuolella löytyi menetelmiä, jotka väitetysti suoriutuvat prosessista kutakuinkin reaaliajassa. Näitä ei ehditty tutkia kovin tarkasti, mutta yksi tehostustapa algoritmille olisi jakaa kiinteät osat omiksi alueikseen. Tämä vähentää huomattavasti vertailuoperaatioita pisteiden välillä, kun uusia pisteitä voidaan verrata samalla kertaa koko kiinteää osaa vasten.

Sylintereiden hakeminen pistedatasta oli yllättävän hankalaa. Kamera osoittautui niin epätarkaksi, että sylinterimallin sädetoleranssin täytyi olla yhtä suuri kuin sylinterin itsensä säteen. Mukaan täytyi laskea myös sylinterin sisäpuolelta kaikki pisteet, ikään kuin letku olisi umpinainen. Tämä teki mahdottomaksi alkuperäisen ajatuksen sylintereiden kokoamisen letkuiksi yhdistämällä sylintereiden akseleiden päätepisteet lähellä oleviin mui-

den sylintereiden päätepisteisiin. Kuvassa 6.3a on nähtävissä tämä alkuperäinen ajatus, kun taas kuvassa 6.3b nähdään millaiseksi sylinteridata muodostui käytännössä. Sylinterit ovat pidempiä kuin etukäteen arveltiin ja niitä on useita sisäkkäin. Se monimutkaistaa hieman letkufunktion muodostamista, mutta tästä huolimatta on mahdollista sijoittaa sylintereiden päätepisteet avaruuteen, jolloin niistä muodostuu letku. Sisäkkäisten sylintereiden käyttäminen jopa helpottaa algoritmin toimintaa, koska yksittäiset virhesylinterit eivät johda algoritmia harhaan. Toisaalta kuten kuvasta 6.4b nähdään, voivat omat ongelmansa aiheuttaa laajat alueet, jotka RANSAC on tulkinnut kasaksi sylintereitä.

(a) Prosessoitava pistepilvi siinä vaiheessa, kun se on erotettu referenssikuvasta



(b) Erotettu pistepilvi sylintereiden etsimisen

Kuva 6.4: Letkuntunnistusalgoritmin vaiheita

6.3 Ratkaisemattomat ongelmat

Selvää on, että prototyyppilaite ei täytä kaikkia sille asetettuja vaatimuksia. Suurin osa tekijöistä liittyy ympäristöön liittyviin haasteisiin ja siihen, miten konenäkö on haastavaa muuttuvissa ja likaisissa olosuhteissa. Silti prototyyppiäkin ajatellen projektissa jäi ratkaisematta tai kokonaan vaille huomiota useita teknisiä haasteita. Ensimmäinen letkun tunnistamiseen liittyvä haaste tuli esille jo projektin alkuvaiheessa, kun maasta tulevien panostusreikien suojaputkien huomattiin muistuttavan merkittävästi panostusletkua niin kooltaan kuin muodoltaan. Varsinaista testidataa ei saatu siitä, miltä

nämä suojaputket näyttävät kameralle. Vielä merkittävämpi ongelma kävi ilmi vasta paljon myöhemmin projektissa, kun ensimmäiset raakadatat saatiin kuvattua tunnelista. Panostusprosessin edetessä jokaisesta reiästä tulee nallin sytytyslanka, joka Kinectille näyttää paljon enemmän letkumaiselta kuin oli oletettu. Tämä johtuu todennäköisesti pääasiassa kameran epätarkkuudesta, jolloin pienikin viiva näyttää kameran toleranssin paksuiselta letkulta. Lankaongelmaan on vaikea keksiä ratkaisua nykyisellä prosessilla. On kuitenkin mahdollista, että paremmalla kuvauslaitteistolla nallit eivät välttämättä aiheuta ongelmaa lainkaan.

Toinen suuri tekijä mihin prototyyppi ei ota kantaa on kameran paikantaminen. Ongelma juontaa juurensa kameran pieneen kuvakulmaan, mikä tarkoittaa, että koko panostettava seinämä ei ole näkyvissä koko aikaa yhdelle kameralle. Jos oletetaan todellisuuteen nähden laajalla 90 asteen kuvakulmalla varustettu kamera, joka metrin päästä seinästä, voidaan laskea sen kuvaaman alueen leveydeksi noin 2 m. Koko tunnelin panostusleveyden voidaan olettaa olevan 5-10 m, joten algoritmin täytyisi pahimmillaan tunnistaa 4 neliömetrin alue noin 50 neliömetrin alueesta. Suhdetta voidaan hieman parantaa usealla kameralla, joiden sijainnit toisiinsa nähden tunnetaan. Pohdittiin, että ICP algoritmia voisi käyttää löytämään referenssikuvasta sen alueen, johon kamera kullakin hetkellä kohdistuu. Tutkittiin, että algoritmi ei välttämättä suoraan sovellu suhteellisen pienen alueen löytämiseen laajasta seinämästä, vaan se saattaa löytää paikallisen minimin [67]. Toisaalta, jos algoritmille annetaan hyvä alkuarvaus esimerkiksi edellisen sijainnin perusteella, pitäisi sen toimia moitteetta.

Viimeinen ratkaisematon ongelma liittyy referenssikuvan muodostamiseen. Prototyyppi olettaa, että referenssikuva otetaan ennen panostamisen aloittamista. Tässä on kaksi suurta ongelmakohtaa. Ensinnäkin menetelmä vaatii panostajilta ylimääräistä työtä. Heidän täytyy jotenkin pystyä määrittämään seurantalaitteistolle, milloin referenssikuvaa ollaan ottamassa. Tämä vaatii ylimääräistä käyttöliittymää ja näytön, jotta nähdään että koko panostusalue on kuvassa. Toinen ongelma liittyy suhteellisen kaukaa otetun kuvan resoluutioon. Jos oletetaankin, että kamera näkee tarpeeksi kauas koko alueen kuvaamiseksi, tulee seinämän pistepilvestä todella harva. Tämä tarkoittaa, että edellä pohdittu ICP algoritmi ei todennäköisesti löydä vastaavia piirteitä sekä harvasta seinämästä, että tiiviistä lähikuvasta. Tämän vuoksi referenssikuvan muodostuksen automatisoinnin kehittäminen olisi ensiarvoisen tärkeää.

6.4 Konenäkösovellusten käytännöllisyys

Yksi lähtökohta tutkimukselle oli, että mikä tahansa menetelmä valittaisiinkin, tulisi sen olla tehokkaampi kuin käsin merkitseminen. Tällaista menetelmää ei löydetty. Vaikka kehitetty konenäkösovellus olisi toiminut täydellisesti, ei se välttämättä olisi ollut käytännöllinen panostajien käyttöön. Prototyyppiä käyttäen kuvausprosessi olisi muodostunut seuraavista vaiheista:

1. Tietokoneen käynnistäminen
2. Seurantaohjelmiston käynnistäminen
3. Panostuskorin asemoiminen siten, että kamera näkee koko alueen
4. Referenssikuvan ottaminen
5. Panostuskuvion asemoiminen referenssikuvaan
6. Seurannan käynnistäminen
7. Panostaminen normaalisti
8. Seurannan lopettaminen

Monet näistä vaiheista voi ajan kanssa automatisoida. Tietokone voi käynnistyä automaattisesti panostuslaitteen käynnistyessä ja seurantaohjelmisto voi käynnistyä samalla. Teoriassa myös seurantaa voidaan tehdä jatkuvasti sen jälkeen kun referenssikuva on otettu ja jatkaa kunnes laitteisto sammutetaan. Kaikkea ei kuitenkaan voi automatisoida. Panostajille suurin vaiva muodostuu referenssikuvan ottamisesta. Jos referenssikuva muodostuisi panostuksen aikana ja edellämainitut optimoinnit saadaan tehtyä, ei panostajille aiheutuisi ylimääräistä vaivaa lainkaan. Tämä tietysti sillä oletuksella, että laskenta tehdään panostuslaitteistolla eikä tiedonsiirtoon tarvitse käyttää resursseja. Tällä tavalla ainoaksi ihmisen velvollisuudeksi jäisi panostuskuvion yhdistäminen kokonaiskuvaan.

Mainittakoon, että mahdollinen muodostettu referenssikuva voi olla käytännöllinen jälkikäteen muihin tarkoituksiin. Panostuskuviot ovat nykyään valmiiksi toteutettu tietokoneavusteisesti, mutta ne voitaisiin referenssikuvan avulla visualisoida katko kerrallaan. Tarkkaa referenssikuvaa voisi myös käyttää jonkinlaiseen räjähdyssimulaatioon, jos sille joskus ilmenee tarvetta.

6.5 Jatko

Yhteenvetona edellisestä voidaan todeta, että tutkittu menetelmä ei ole käytännöllinen tutkimusongelman ratkaisemiseksi. Pääasiallinen ongelma on ympäristö, jossa mikään ei puolla menetelmän käyttöä. Kosteus, lika ja ahtaus tekevät kameran epäkäytännölliseksi maan alla. Lisäksi käytettyjen kameroiden tarkkuus ei riitä luotettavaan tunnistamiseen. Tästä huolimatta konenäkö vaikuttaa edelleen käytännöllisimmältä vaihtoehdolta, sillä muut menetelmät ovat liian työläitä panostajille tai niiden tarkkuuden ei uskota riittävän. Jos konennäkösovellusta kuitenkin halutaan kehittää eteenpäin, on ensimmäisen jatkokehityskohteen oltava toimivan kamerakonfiguraation löytäminen. Tavoitteena tulee silloin olla järjestely, jossa kamerat näkevät koko prosessin luotettavasti koko ajan. Toinen mahdollinen kehitysvaihtoehto voisi myös olla kamerasovelluksen yhdistäminen johonkin muuhun menetelmään. Tällöin toinen menetelmä voi paikata konenäköä silloin kun konenäkö ei pysty paikantamaan letkua.

Kaiken edellämainitun pohjalta muodostettu johtopäätös on, että käsin kirjaaminen on edelleen tehokkain ja luotettavin vaihtoehto reikien panostusjärjestyksen seuraamiseen. Vaikka konenäkösovellukset ovat kevyemmästä päästä mitä tulee vaivannäköön, eivät ne kuitenkaan ole täysin vaivattomia. Haluttu räjäytyskuvio täytyy joka tapauksessa valita, kuten täytyy myös suorittaa räjäytyskuvion yhdistäminen referenssikuvaan. Panostajan kannalta käsinkirjausprosessi voi tuntua vaivalloiselta, mutta toisaalta varta vasten suunnitellulla käyttöliittymällä ja tunneliympäristöä varten rakennetulla laitteistolla se voi olla käytännöllisin vaihtoehto.

Luku 7

Yhteenveto

Tämän diplomityön tutkimusongelma oli selvittää, mikä on paras tapa emulsiopanostusprosessin seuraamiseen. Tarkemmin ottaen kiinnostava kysymys on, kuinka paljon emulsiopanostusprosessissa käytetään emulsiota kutakin porattua reikää kohden. Alue rajattiin tunneliympäristöön, minkä vuoksi käsiteltiin tunnelinlouhintaprosessia ensiksi kokonaisuutena. Varsinainen panostusprosessi kuvattiin sen jälkeen suuremmalla tarkkuudella. Työssä analysoitiin myös tunneliympäristön rajoitteita, joista merkittäviksi osoittautuivat likaisuuden ja kosteuden lisäksi erityisesti tunneliympäristön tilanpuute - tunneliin on vaikea mahduttaa ylimääräisiä sensoreita niin, että ne voivat luotettavasti tarkkailla koko prosessia ilman katvealueita.

Vaihtoehtoisia menetelmiä esitettiin viisi, joista neljä voisi paremmissa olosuhteissa ja pienemmillä tarkkuusvaatimuksilla toimia ongelman ratkaisuna. Nämä neljä menetelmää olivat kuvasta tunnistus, muodon tunnistaminen pistepilvestä, letkun seuraaminen radioaaltojen avulla ja IMU:n hyväksikäyttö. Jokaisen hyviä ja huonoja puolia tutkittiin ja tämän perusteella valittiin yksi menetelmä prototyyppitoteutusta varten. Kustannusten vuoksi ja panostajan vaiva minimoiden päädyttiin prototyyppi toteuttamaan pistepilvien avulla. Perusideana tässä ja muissa menetelmissä oli selvittää letkun sijainti tunnetun poraus-räjäytyskaavion suhteen. Tämä tieto voidaan yhdistää myöhemmin panostuslaitteelta saatuun dataan kulutetun emulsion määristä panostushetkillä, jolloin saadaan tietää kokonaiskulutus reikää kohden.

Prototyyppitoteutusta varten tutkittiin erilaisia pistepilvimenetelmiä, joista lopulta valittiin RANSAC -algoritmi letkun tunnistamiseen. Prototyyppi rakennettiin siihen vaiheeseen, että se osasi erottaa taustan liikkuvista muodoista ja tunnistaa letkun osat. Letkun osien avulla voidaan sopivalla heuristiikalla päätellä, mitkä havaituista palasista kuuluvat panostusletkuun ja mitkä ovat virhehavaintoja esimerkiksi panostajan käsivarresta. Muita

virhehavaintoja voi tulla panostuksen aikana esimerkiksi reikiin pistettävistä sytytinjohtimista.

Prototyyppitoteutus olettaa, että sitä varten on otettu referenssikuva ja että kamera pysyy paikoillaan kuvan ottamisen jälkeen. Käytännössä toimiva laite vaatisi menetelmän, jolla referenssikuva voidaan tuottaa panostuksen aikana liikkuvalla kameralla. Näin saadaan parempi pistetiheys tunnistusta varten eikä panostajan tarvitse käyttää vaivaa referenssikuvan ottamiseen.

Kameraksi prototyyppiä varten valittiin Kinect 2.0 3D-kamera, koska se on markkinoiden hintatehokkain ratkaisu pistepilvien tuottamiseen. Kamerasta havaittiin, että sen täytyy antaa lämmitä noin 20 minuuttia, ennen kuin sen tulokset stabiloituvat. Merkittävämpi tulos on kuitenkin se, että kameran tuottamista pistepilvistä oli vaikea löytää sylintereitä sen liian heikosta tarkkuudesta johtuen. RANSAC:n sylinterimallin sädetoleranssi täytyi asettaa letkun säteeseen nähden hyvin suureksi, ennen kuin algoritmi alkoi löytää oikeita sylintereitä. Viimeinen kameraan liittyvä tulos koskee kuvausaluetta ja sijoituspaikkoja. Kameran kuvakulma on liian pieni, että se voisi nähdä koko kuvausalueen yhdellä kertaa panostuskoriin liitettynä. Toisaalta sivulle asetetulla kameralla tulee kuvausalueelle kielekkeitä ja kameran erotuskyky heikkenee, kun reiät ovat kuvakulmasta katsoen lähekkäin toisiaan. Ainoa käytännöllinen tapa seurata prosessia on sijoittaa useita kameroita panostuskoriin siten, että ne näkevät koko panostusalueen kerralla. Todennäköisesti ainakin jokaista reikää tarvitsee seurata useasta kuvakulmasta, sillä panostaja liikkuu helposti reikien eteen.

Loppupäätelmä on, että pistepilvimenetelmä ei todennäköisesti ole yksin tarpeeksi luotettava menetelmä panostuksen seuraamiseksi. Suurin syy tälle on näkyvyyden estyminen. Yksi jatkokehityssuunta onkin tutkia, voidaanko löytää sellaista kamerakonfiguraatiota, jolla koko panostusprosessi on seurattavissa. Toinen vaihtoehto on käyttää jonkinlaista yhdistelmäsensoria, jossa konenäköä tuetaan jollakin toisella sensorityypillä. Tämän diplomityön pohjalta näyttää kuitenkin siltä, että tehokkain tapa seurata panostusprosessia on kirjata panostettavat reiät käsin suoraan elektroniseen järjestelmään.

Kirjallisuutta

[1] Per-Anders Persson, Roger Holmberg, and Jaimin Lee. *Rock blasting and explosives engineering*. CRC press, 1993.

[2] Umit Ozer. Environmental impacts of ground vibration induced by blasting at different rock units on the kadikoy–kartal metro tunnel. *Engineering geology*, 100(1):82–90, 2008.

[3] Cengiz Kuzu. The mitigation of the vibration effects caused by tunnel blasts in urban areas: a case study in istanbul. *Environmental geology*, 54(5):1075–1080, 2008.

[4] Valtioneuvosto. Valtioneuvoston asetus räjäytys- ja louhintatyön turvallisuudesta (644/2011), 16 kesäkuuta 2011. `http://www.finlex.fi/fi/laki/alkup/2011/20110644`.

[5] Valtioneuvosto. Valtioneuvoston asetus räjähteiden valmistuksen, käsittelyn ja varastoinnin turvallisuusvaatimuksista (1101/2015), 1 syyskuuta 2015. `http://www.finlex.fi/fi/laki/alkup/2015/20151101`.

[6] Qing Chen, Hong-Tu Wang, Guo-Zhong Hu, Xiao-Hong Li, Kai-Xue Li, and Cheng Pang. Monitoring and controlling technology for blasting vibration induced by tunnel excavation. *ROCK AND SOIL MECHANICS-WUHAN-*, 26(6):964, 2005.

[7] Wenbo Lu, Jianhua Yang, Peng Yan, Ming Chen, Chuangbing Zhou, Yi Luo, and Li Jin. Dynamic response of rock mass induced by the transient release of in-situ stress. *International Journal of Rock Mechanics and Mining Sciences*, 53:129–141, 2012.

[8] Joseph Egan, Jeff Kermode, Martin Skyrman, and Loren L Turner. Ground vibration monitoring for construction blasting in urban areas. *Final Report, Caltrans*, pages 1–11, 2001.

[9] Länsimetro. Louhintavaiheet. `https://www.lansimetro.fi/rakentaminen/louhintavaiheet/`, käyty 7.11.2017.

[10] Jean Paul Godard. Urban underground space and benefits of going underground. In *World tunnel congress*, 2004.

[11] Tommi Vuolio, Raimo ja Halonen. *Räjäytystyöt, Luku 1. Räjähteiden ja louhintatekniikan kehitys*. Suomen Rakennusmedia, 2012.

[12] Tommi Vuolio, Raimo ja Halonen. *Räjäytystyöt, Luku 7. Maanalainen louhinta*. Suomen Rakennusmedia, 2012.

[13] Thomas R Kuesel, Elwyn H King, and John O Bickel. *Tunnel engineering handbook*. Springer Science & Business Media, 2012.

[14] Simon Emsley, Olle Olsson, L Stenberg, HJ Alheid, and S Falls. Zedex-a study of damage and disturbance from tunnel excavation by blasting and tunnel boring. Technical report, Swedish Nuclear Fuel and Waste Management Co., 1997.

[15] Tommi Vuolio, Raimo ja Halonen. *Räjäytystyöt, Luku 2. Räjähdysaineet*. Suomen Rakennusmedia, 2012.

[16] Rafael Fernandez-Rubio and David Fernandez Lorca. Mine water drainage. *Mine water and the environment*, 12(1):107–130, 1993.

[17] Risto Hienonen and Reima Lahtinen. *Korroosio ja ilmastolliset vaikutukset elektroniikassa*. Valtion teknillinen tutkimuskeskus, 2000.

[18] C Shawn Green and Daphne Bavelier. Exercising your brain: a review of human brain plasticity and training-induced learning. *Psychology and aging*, 23:692, 2008.

[19] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (2nd Ed)*. Prentice Hall, 2002.

[20] Stan Birchfield. Elliptical head tracking using intensity gradients and color histograms. In *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, pages 232–237. IEEE, 1998.

[21] Gregory D Hager and Peter N Belhumeur. Real-time tracking of image regions with changes in geometry and illumination. In *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR'96, 1996 IEEE Computer Society Conference on*, pages 403–410. IEEE, 1996.

[22] Valtteri Takala and Matti Pietikainen. Multi-object tracking using color, texture and motion. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–7. IEEE, 2007.

[23] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2001.

[24] Satya Mallick. Image recognition and object detection : Part 1. `https://www.learnopencv.com/image-recognition-and-object-detection-part1/`, käyty 7.11.2017.

[25] Liang Zhao and Charles E Thorpe. Stereo-and neural network-based pedestrian detection. *IEEE Transactions on Intelligent Transportation Systems*, 1(3):148–154, 2000.

[26] Ian H Witten, Eibe Frank, Mark A Hall, and Christopher J Pal. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[27] Roger Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal on Robotics and Automation*, 3(4):323–344, 1987.

[28] Hedvig Sidenbladh, Michael J Black, and David J Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *European conference on computer vision*, pages 702–718. Springer, 2000.

[29] Christoph Bregler, Aaron Hertzmann, and Henning Biermann. Recovering non-rigid 3d shape from image streams. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 690–696. IEEE, 2000.

[30] Ruwen Schnabel, Roland Wahl, and Reinhard Klein. Efficient ransac for point-cloud shape detection. In *Computer graphics forum*, volume 26, pages 214–226. Wiley Online Library, 2007.

[31] Georgios Florides and Soteris Kalogirou. Annual ground temperature measurements at various depths. In *8th REHVA World Congress*, 2005.

[32] Dragos Niculescu and Badri Nath. Ad hoc positioning system (aps). In *Global Telecommunications Conference, 2001. GLOBECOM'01. IEEE*, volume 5, pages 2926–2931. IEEE, 2001.

[33] Chris Savarese, Jan M Rabaey, and Jan Beutel. Location in distributed ad-hoc wireless sensor networks. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, volume 4, pages 2037–2040. IEEE, 2001.

[34] Kamin Whitehouse, Chris Karlof, and David Culler. A practical evaluation of radio signal strength for ranging-based localization. *ACM SIGMOBILE Mobile Computing and Communications Review*, 11(1):41–52, 2007.

[35] Manikanta Kotaru, Kiran Joshi, Dinesh Bharadia, and Sachin Katti. Spotfi: Decimeter level localization using wifi. In *ACM SIGCOMM Computer Communication Review*, volume 45, pages 269–282. ACM, 2015.

[36] Isaac Skog and Peter Händel. Calibration of a mems inertial measurement unit. In *XVII IMEKO World Congress*, pages 1–6, 2006.

[37] Antonio R Jimenez, Fernando Seco, Carlos Prieto, and Jorge Guevara. A comparison of pedestrian dead-reckoning algorithms using a low-cost mems imu. In *Intelligent Signal Processing, 2009. WISP 2009. IEEE International Symposium on*, pages 37–42. IEEE, 2009.

[38] Forcit. Käyttöturvallisuustiedote; kemiitti 510 matriisi, kemiitti 610 matriisi, kemiitti 810 matriisi. `http://forcit.fi/assets/product-brochures/KTT-Kemiitti-510-610-810-matriisi-25.5.2015.pdf`, käyty 12.11.2017.

[39] Käyttäjänimi Btyner. Kuva kolmiulotteisesta kd-puusta. `https://en.wikipedia.org/wiki/K-d_tree#/media/File:3dtree.png`.

[40] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

[41] Michael Connor and Piyush Kumar. Fast construction of k-nearest neighbor graphs for point clouds. *IEEE transactions on visualization and computer graphics*, 16(4):599–608, 2010.

[42] Sergio Orts-Escolano, Vicente Morell, Jose Garcia-Rodriguez, and Miguel Cazorla. Point cloud data filtering and downsampling using growing neural gas. In *Neural Networks (IJCNN), The 2013 International Joint Conference on*, pages 1–8. IEEE, 2013.

[43] M Alexa, S Rusinkiewicz, Diego Nehab, and Philip Shilane. Stratified point sampling of 3d models. In *Proc. Eurographics Symp. on Point-Based Graphics*, pages 49–56, 2004.

[44] Radu Bogdan Rusu. Semantic 3d object maps for everyday manipulation in human living environments. *KI-Künstliche Intelligenz*, 24(4):345–348, 2010.

[45] Hough Paul VC. Method and means for recognizing complex patterns, December 18 1962. US Patent 3,069,654.

[46] Robert C Bolles and Martin A Fischler. A ransac-based approach to model fitting and its application to finding cylinders in range data. In *IJCAI*, volume 1981, pages 637–643, 1981.

[47] Sunglok Choi, Taemin Kim, and Wonpil Yu. Performance evaluation of ransac family. *Journal of Computer Vision*, 24(3):271–300, 1997.

[48] Tahir Rabbani and Frank Van Den Heuvel. Efficient hough transform for automatic detection of cylinders in point clouds. *Isprs Wg Iii/3, Iii/4*, 3:60–65, 2005.

[49] Pointclouds.org documentation. Cylinder model segmentation. `http://pointclouds.org/documentation/tutorials/cylinder_segmentation.php`, käyty 12.11.2017.

[50] Paul J Besl, Neil D McKay, et al. A method for registration of 3-d shapes. *IEEE Transactions on pattern analysis and machine intelligence*, 14(2):239–256, 1992.

[51] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568. ACM, 2011.

[52] Larry Li. Time-of-flight camera–an introduction. *Technical white paper*, (SLOA190B), 2014.

[53] Thomas Algie Abrams Jr. Stereoscopic video, January 15 2008. US Patent 7,319,720.

[54] CNET. Timeline: A look back at kinect's history. `https://www.cnet.com/news/timeline-a-look-back-at-kinects-history/`, käyty 11.11.2017.

[55] François Blais. Review of 20 years of range sensor development. *Journal of electronic imaging*, 13(1):231–243, 2004.

[56] Zhengyou Zhang. Microsoft kinect sensor and its effect. *IEEE multimedia*, 19(2):4–10, 2012.

[57] Olaf Hall-Holt and Szymon Rusinkiewicz. Stripe boundary codes for real-time structured-light range scanning of moving objects. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 359–366. IEEE, 2001.

[58] Microsoft. Kinect hardware key features and benefits. `https://developer.microsoft.com/en-us/windows/kinect/hardware`, käyty 11.11.2017.

[59] Structure. Structure sensor technical specifications. `https://structure.io/support/what-are-the-structure-sensors-technical-specifications`, käyty 11.11.2017.

[60] Structure. Structure core technical specifications.

[61] Oliver Wasenmüller and Didier Stricker. Comparison of kinect v1 and v2 depth images in terms of accuracy and precision. In *Asian Conference on Computer Vision*, pages 34–45. Springer, 2016.

[62] Verkkokauppa.com. Microsoft kinect-sensori, pc / xbox one. Hinta 2.5.2017.

[63] Structure. Structure sensor. Hinta 2.5.2017.

[64] Teknikum. 2017 letkut. `http://teknikum-com-bin.aldone.fi/@Bin/c768dd3a24ed285b6cbeb2b5465fb7fa/1510446061/application/pdf/308816/Teknikum-hoses-2017.pdf`, käyty 12.11.2017.

[65] JC Ralston, CO Hargrave, and DW Hainsworth. Localisation of mobile underground mining equipment using wireless ethernet. In *Industry Applications Conference, 2005. Fourtieth IAS Annual Meeting. Conference Record of the 2005*, volume 1, pages 225–230. IEEE, 2005.

[66] Pietro Zanuttigh, Giulio Marin, Carlo Dal Mutto, Fabio Dominio, Ludovico Minto, and Guido Maria Cortelazzo. *Operating Principles of Time-of-Flight Depth Cameras*, pages 81–113. Springer International Publishing, Cham, 2016.

[67] Dmitry Chetverikov, Dmitry Svirko, Dmitry Stepanov, and Pavel Krsek. The trimmed iterative closest point algorithm. In *Pattern Recognition,*

2002. Proceedings. 16th International Conference on, volume 3, pages 545–548. IEEE, 2002.